NR INDEKSU 353965
PL ISSN 0860-1674

3

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 3(27) MARZEC 1988 CENA 150 ZŁ



KOMPUTER '88

JUŻ PISZĘ!

PĘTLA INTERPRETERA

FIRMA BEZ GRANIC

XIO

GRYZIENIE ORZECHÓW

Co jest najważniejsze w całej tej sympatycznej zabawie pod nazwą „informatyka"? Część z naszych Czytelników — rekrutująca się z grupy tych, którzy nie mają jeszcze komputerów — odpowie zapewne, że dla nich najważniejszy jest właśnie on, komputer. Ale ci, którzy mają już swojego Atari, Spectrum czy Commodore odkrzykną bez wahania: oprogramowanie!

Pół biedy, gdy zapominają o roli oprogramowania 12-latki, molestujące rodziców o sprawienie im drogiego prezentu, będącego ostatnio symbolem nowoczesności. Ale gorzej, gdy troska o hardware przesłania horyzont instytucjom (a mówiąc precyzyjniej — ludziom w nich zatrudnionym) powołanym do zajmowania się upowszechnianiem wiedzy informatycznej. Mam oczywiście na myśli szkołę.

Większość dotychczasowych publicznych dyskusji na temat informatyki w szkole zdominowana była przez problemy sprzętowe: jaki powinien być szkolny komputer i skąd go wziąć? Na pytanie to odpowiadano różnie — wyliczano optymalne pojemności pamięci, standarty, przywoływano patriotyzm i racje naszego przemysłu elektronicznego... Dużo emocji wzbudził konkurs na polski komputer „szkolny". Jeszcze więcej emocji ujawniło się przy dyskusji o ewentualnych zakupach za granicą... Praktyka tymczasem kierowała się swoją logiką, w nielicznych tylko punktach przystającą do tych namiętnych dyskusji.

Niezależnie bowiem od wszystkich programowych, „strategicznych" ustaleń coraz więcej szkół zaczęło mieć w swych klasach komputery. Pochodziły one z różnych źródeł: 300 sztuk „Spectrum" przekazała polskim szkołom w darze austriacka firma „Reiter", sporo przyzwoitych IBM-podobnych komputerów dała szkołom rodzima „Agrotechnika", do paru placówek szkolnych trafiły z Zabrza maszyny marki „Meritum", sporo sprzętu zakupiły do szkół instytucje patronackie — zakłady przemysłowe, wojsko, rady narodowe (przykład z Warszawy: rady narodowe Śródmieścia i Żoliborza wysupłały na sprzęt komputerowy dla swoich szkół po 25 mln złotych!)... Oczywiście, gwoli prawdzie należy dodać, że część komputerów trafiła do szkół również poprzez ministerstwo edukacji.

I oto w pewnym momencie okazało się, że wprawdzie dyskusje o nauczaniu informatyki w szkołach trwają w najlepsze, ale jednocześnie jest już w nich na tyle dużo komputerów, aby zacząć się zastanawiać nad właściwym ich wykorzystaniem. Wyszło wówczas na jaw to, co naprawdę stanowi barierę szkolnej edukacji informatycznej! Pisze o tym w ciekawym artykule zamieszczonym na łamach „Życia Gospodarczego" red. Krzysztof Fronczak.

Hamulcem rozwoju informatyki w szkołach — stwierdza Fronczak — nie jest niedobór środków technicznych. Brakuje przygotowanych merytorycznie nauczycieli i... nauczycieli nauczycieli. Pedagogów chcących dokształcać się w informatyce jest więcej niż fachowców od informatyki chcących im w tym pomóc. Śmieszne warunki płacowe nie przyciągają jednak najbardziej wartościowej kadry. Z drugiej zaś strony metodyka nauczania informatyki nadal jest w powijakach, mało jest po prostu takich, którzy by się na tym rzeczywiście znali.

Gdyby Ministerstwo Edukacji Narodowej zechciało — to może tę barierę pokonać. Widziałem niedawno jak to się robi w Związku Radzieckim. Będąc w słynnym Uniwersytecie Nowosybirskim spotkałem się z grupą nauczycielek matematyki i fizyki z całej Syberii, które przyjechały na kurs informatyczny. Zdobycie nowej wiedzy było dla niej dużym wysiłkiem, ale — jak dowiedziałem się od rektora tej uczelni profesora Junija Jerszowa — większość kończyła te kursy z wyróżnieniem. Mogą w trybie ekspresowym przygotowywać swoje kadry informatyczne Rosjanie, więc pewnie i my potrafilibyśmy to zrobić. Ale w tym momencie wyłania się kolejna bariera, której pokonanie jest o wiele trudniejsze. Tą drugą zasadniczą barierą są braki w oprogramowaniu. Nie jest to oczywiście żadne odkrycie dla osób znających się choć trochę na informatyce, ale ludzie z resortu edukacji jakby zapomnieli, że informatyka to przede wszystkim programy, programy i jeszcze raz programy! Znalezienie pieniędzy na zakup komputerów to zaledwie jedna setna problemu (optymistycznie licząc!). Schody zaczynają się przy próbie sensownego, edukacyjnego ich zastosowania. Żeby te schody pokonać — trzeba zainwestować. Innego wyjścia nie ma.

Powtórzę za red. Fronczakiem, że nie brak w naszym kraju dobrych programistów potrafiących przygotować sensowne programy edukacyjne. Wymaga to jednak tygodni i miesięcy czasu. Do wysiłku nie zachęcają zaś ani stawki wynagrodzenia, ani pirackie reguły walki na rynku. Łatwiej przecież skopiować obcy program, trochę go „udoskonalić" czy zmienić i oferować za grube pieniądze, niż głowić się samodzielnie, zwłaszcza że owoce tej pracy mogą zostać bezkarnie przez kogoś przywłaszczone... Stworzenie własnej bazy oprogramowania to bez wątpienia jeden z twardszych orzechów do zgryzienia dla władz oświatowych.

Zgryźć go jednak trzeba. Nikt inny tego przecież za nas nie zrobi. Bo i jaki miałby w tym interes?

Waldemar Siwiński



WYBIERZ SAM

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05

Przewodniczący Rady Redakcyjnej: Jerzy Domański-redaktor naczelny „Sztandaru Młodych"

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Roman Poznański (z-ca sekretarza redakcji „SM" — sekretarz zespołu „Bajtka"), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze Leopold Dzikowski.

Klany redagują:
Commodore — Klaudiusz Dybowski,
Amstrad-Schneider — Tomasz Pyć, Sergiusz Wolicki,
Spectrum — Marcin Przasnyski, Michał Szuniewicz,
Atari — Wiesław Migut, Wojciech Zientara.

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Woltańska

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa Telefony: Centrala 13-20-40 do 49 Redakcja Reklamy 13-20-40 do 49 w. 403, 414

Cena 100 zł.

Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51. Nr zlecenia 010188 n. 150.000 egz. U-113



Bajtek

SZANOWNY PANIE REDAKTORZE!

Nazywam się Witold Wiankowski, mam 16 lat.

Komputerami zainteresowałem się pod koniec zeszłego roku, dzięki byłemu, niestety, koledze ze szkoły. Czytelnikiem „Bajtka" jestem od stycznia br., ale znam Wasze pismo od numeru 2/86. Pozwolę sobie podzielić się z Wami kilkoma uwagami na jego temat. Oto one:

Sporo do życzenia pozostawia ciągle jakość niektórych listingów. Mam słaby wzrok i ich odczytanie sprawia mi niekiedy naprawdę duże trudności (np. „Kasetowy program operacyjny" — Klan Atari nr 5/87). Myślę, że przynajmniej krótkie, a ułatwiające życie programy można by drukować zwykłą czcionką.

Jak wiadomo, „Bajtek" nie może się (niestety!) poszczycić zbyt dużą objętością. Tym bardziej dziwi marnotrawstwo miejsca przeznaczonego pod ilustracje, które z powodzeniem mogłyby być 2-3 razy mniejsze, ew. całkowicie zniknąć (np. mapy do gier z powodzeniem zmieściłyby się na jednej stronie). Bardzo denerwujące są też cało — albo nawet dwustronicowe (!) reklamy.

Przeglądając poszczególne numery „Bajtka" rzuca się w oczy, jak dużo miejsca przeznaczacie na gry. A przecież komputer ma również inne zastosowania. Niech publikacje w rodzaju „GEOS — czyli C-64 goni McIntosh'a" (chwała Wam za nią) nie będą rzadkością. Proponuję prezentowanie adaptacji „poważniejszych" programów na komputery dobrze u nas znane. Sam np. z chęcią przekonałbym się, jak prezentuje się w porównaniu z pierwowzorem AutoCad na CPC 6128.

Proponuję częstsze drukowanie klanu nietypowych i prezentowanie tam dobrych, choć niekoniecznie popularnych u nas mikrokomputerów. Nie każdemu przecież potrzebny jest sprzęt z biblioteką -set programów w kraju. Dziwi mnie, że przegapiliście ukazanie się na rynku komputerów Acorn, Bondwell, Olivetti i SpectraVideo — naprawdę przydałyby się ich rzetelne oceny.

Na rynku jest w chwili obecnej b. dużo ciekawych komputerów Sharp MZ-700/700. Maszyna ta ma już tak wielu użytkowników (zaznaczam, że nie należę do ich grona), że dobrze było by otworzyć jej klan. Mógłby się on ukazywać np. na przemian z klanem Meritum.

W swoich zbiorach posiadam wszytkie „Bajtki" od n-ru 2/86. Chciałbym kupić wcześniejsze, ale ich po prostu nie ma. Czy nie dałoby się ich wznowić?

Sprawa testów. Do tej pory ukazały się testy Spectrum 128 +2, Gemini 10x i Timex'a 2048. Czy nie można w każdym numerze umieścić przynajmniej jednego testu? Można by w ten sposób rozwiać wiele mitów o „cudownych" komputerach.

Swego czasu ukazywały się w „Bajtku" wizytówki zachodnich pism komputerowych, wraz z ich adresami. Dlaczego zaprzestaliście ich drukowania? Były naprawdę bardzo pożyteczne.

Proponuję wydrukowanie listy adresów do bardziej znanych firm hardware'owych i software'owych. Umożliwiłoby to, być może, wielu hobbystom dostęp do wielu ciekawych informacji.

Bardzo zdziwiła mnie korespondencja z CEBIT'87 (nr 5/87). Impreza ta jest jedną z największych w świecie, a poświęcono jej 1 2/3 strony! W dodatku autor przez dużą część artykułu „leje wodę" (co może obchodzić czytelnika, ile kosztuje pokój w hotelu, czy prywatna kwatera?). Relacja taka, miast zaciekawić, wzbudza tylko zniecierpliwienie. W dodatku zilustrowana została... trzema fotografiami. Wstyd!

Proponuję wydrukowanie w numerze 12/87 szerokiej ankiety z propozycjami konkretnych zmian. Przypuszczam, że mogłoby to dać bardzo interesujący (również dla czytelników) plon.

Mam nadzieję, że moje uwagi przyczynią się choć w minimalnym stopniu do jeszcze większego „uciekawienia" Waszego pisma. Łączę życzenia dalszej, owocnej pracy.

Z poważaniem
Witold Wiankowski



JESTEŚMY FIRMĄ BEZ GRANIC

RYNEK POLSKI, W PORÓWNANIU DO ZACHODNIOEUROPEJSKIEGO, JEST MAŁY, ALE Z PEWNOŚCIĄ WAŻNY

Rozmowa z Gaudenzem M. Juon, dyrektorem marketingu i sprzedaży koncernu „Star" na Europę, Bliski Wschód i Afrykę oraz Christophe J. Musiałem, dyrektorem handlowym „Stara" na Europę.

— Czy „Star" to koncern japoński, czy już ponadnarodowy?

— „Star" powstał przed 40 laty jako firma japońska. Założyciel, pan Seiichi Sato jest do dziś prezydentem.

— I produkcja, przynajmniej drukarek, prowadzona jest także tylko w Japonii?

— Jak na razie tak. Od marca br. rozpocznie jednak produkcję także fabryka drukarek „Star" w Walii. Wytwarzać się tam będzie około 30 tys. sztuk miesięcznie. Natomiast cała produkcja firmy to prawie milion drukarek w 1987 roku.

— Żeby je sprzedać potrzeba profesjonalnego marketingu i wielkiego systemu sprzedaży. Jak są one zorganizowane?

— Produkcja, jak już powiedzieliśmy, umiejscowiona jest w Japonii, chociaż część zakładów kooperujących znajduje się też w Korei Południowej. Następnym ogniwem firmy są jej oddziały i wielkie biura sprzedaży. Mamy takie biuro w Stanach Zjednoczonych, Wielkiej Brytanii, Hongkongu i w Australii. To, w którym my pracujemy, usytuowane jest w RFN i obsługuje Europę, Bliski Wschód i Afrykę. W grudniu 1987 roku powstało także podobne biuro we Francji podległe jednak naszemu oddziałowi.

Każdy oddział ustala swoją strategię marketingu. W naszym przypadku opiera się ona na rozwoju sieci dystrybutorów. W jednym kraju mamy ich czasem kilku, a czasem tylko jednego. Przeważają jednak ci z wyłącznością na dany kraj.

— A jakie są różnice pomiędzy handlem i marketingiem prowadzonymi tu w Europie i w Stanach Zjednoczonych?

— Nie możemy zbyt wiele powiedzieć o tym, jak wygląda marketing „Stara" na rynku amerykańskim. Wiemy, że towar trafia tam do klienta poprzez trzy lub cztery różne kanały sprzedaży. U nas w Europie, jak wspomnieliśmy, przeważają dystrybutorzy narodowi lub, jak w RFN, sieć dealerów. Mamy ich tam około 200. We Francji jest to zorganizowane w podobny sposób. W innych krajach, oprócz Polski, pracujemy z dystrybutorem, który sprzedaje nasz produkt poprzez całą sieć dealerów. W przypadku waszego kraju jest nieco inaczej. Mamy tu „ABC Data" — w zasadzie też dystrybutora. Jednak rozpowszechnianie zachodzi na nieco innym poziomie — ABC „Data" sprzedaje i indywidualnym klientom i pośrednikom. Dzięki temu ceny są niższe niż w innych krajach Europy.

— Na razie jednak w ABC „DATA" wasze drukarki kupić może tylko ten, kto posiada dewizy na koncie. Czy nie myślicie także o innych formach eksportu do naszego kraju poprzez wymianę kompensacyjną, joint ventures ltd.?

— „ABC Data" już pracuje w różnych formach na rynku, na którym jest dystrybutorem. Kompensacja, inne formy przychodzą wraz z rozwojem firmy. A ten rozwój jest widoczny. Już podczas targów Komputer'88, mówiło się sporo o nowych formach współpracy. Nie chciałbym jednak więcej o tym mówić choćby dlatego, że przecież „ABC Data" jest samodzielną firmą działającą na własne konto.

— Zostawmy na chwilę handel i zajmijmy się techniką. Na rynku mikrokomputerów profesjonalnych mamy w zasadzie do czynienia ze standardem wyznaczonym przez IBM. Dlaczego takiego standardu nie ma wśród drukarek?

— Trzeba byłoby najpierw ustalić co to znaczy standard. Drukarki to tylko urządzenia peryferyjne komputerów i właśnie do nich się je dostosowuje, a nie odwrotnie. Oczywiście, gdy wziąć pod uwagę szczegóły, jest wiele różnic pomiędzy poszczególnymi markami. Firmy japońskie utworzyły komisję, której zadaniem byłoby zbadanie możliwości zbliżenia różnych konstrukcji, stworzenia czegoś w rodzaju standardu.

— Na razie jednak tendencje są różne, różna polityka i różne efekty. Jak w tej konkurencji plasuje się dziś „Star"?

— W Europie z pewnością jesteśmy jednymi z pierwszych. Jest kilka krajów, w których rynek drukarek został przez nas praktycznie zdominowany.

— Na przykład w Polsce?

— Nie tylko. Przodujemy także na rynku RFN, austriackim, greckim, czy krajów Beneluksu. Generalnie w Europie pozycja „Stara" jest bardzo dobra, lepsza niż np. w Stanach Zjednoczonych. Liczba naszych drukarek sprzedawanych w USA jest wprawdzie większa niż na Starym Kontynencie, wynika to jednak tylko z wielkości rynku.

— Jakie, według Panów zalety drukarek „Stara" decydują o takiej ich popularności?

— Sukces naszej firmy bazuje przede wszystkim na dynamicznej polityce marketingu, na niezwykle elastycznej strategii cenowej, a także oczywiście na jakości technicznej produktów i wysokim poziomie oferowanego serwisu.

Oferujemy wyroby bardzo nowoczesne. Dlatego też musimy nieustannie szkolić naszych sprzedawców, organizować im specjalne kursy. Dystrybutorzy nasi są maksymalnie elastyczni, uwzględniają w działaniu specyfikę swego rynku.

— A który z modeli oferowanych przez firmę polecilibyście Panowie dziś czytelnikom „Bajtka"?

— Z tych, które pokazaliśmy na wystawie Komputer'88 oczywiście LC-10, którą to drukarkę określić można bez przesady jako produkt rewolucyjny pod względem technicznym, a przy tym stosunkowo niedrogi. W porównaniu do NL-10 nowa drukarka ma wiele ulepszeń. Między innymi osiem wbudowanych krojów pisma i podajnik pojedynczych stron papieru.

Inna, niezwykle udana nasza konstrukcja, której nie pokazaliśmy jeszcze w Warszawie to LC-10 Colour. Jest to chyba wyrób bez konkurencji — tania drukarka kolorowa. Można w niej stosować także kasetę czarną z normalnej LC-10. W ten sposób za bardzo małą dopłatą uzyskuje się potencjalną możliwość przekształcenia w każdej chwili drukarki jednobarwnej w wielobarwną tylko poprzez zmianę kasety.

— A co polecacie bardziej wymagającym klientom-profesjonalistom?

— Drukarkę laserową oraz nasze drukarki 24-igłowe: NL 24-10/15 i NB-15. W połowie roku do naszej oferty dołączy także zapewne nowa szybka drukarka 24-igłowa.

— Czy nowości te oznaczają ostateczne rozstanie z tak popularną u nas SG-15 cenioną choćby ze względu na możliwość użycia taśmy maszynowej?

— Musimy już z niej zrezygnować. Produkcja tej drukarki jest zbyt kosztowna. W ubiegłym roku utrzymaliśmy ją tylko ze względu na zamówienia z Polski. A co do kaset, to wydaje nam się, że już nie ma z nimi problemu. Te do LC-10 będą zresztą dużo tańsze od kaset NL-10 przy nieco tylko obniżonej trwałości. Ponieważ mamy serwis w Warszawie, nie przewidujemy także kłopotów z dostępem do nowych kaset.

— Utrzymanie przez rok produkcji jednego typu drukarki tylko ze względu na nasz rynek świadczyłoby o tym, że traktujecie poważnie nawet niewielkich kontrahentów.

— Rynek polski, w porównaniu do zachodnioeuropejskiego, jest mały, ale z pewnością ważny. Przypuszczamy, że w 1988 roku sprzedamy wam co najmniej 20 tysięcy naszych drukarek i, co chyba najważniejsze, sprzedaż nasza znacznie się powiększa. Byliśmy dwa lata temu na pierwszych w waszym kraju Targach komputerowych w hotelu „Viktoria". Zapamiętaliśmy te tłumy ściśnięte na małej powierzchni wystawy. Tegoroczne targi dysponowały już znacznie większą powierzchnią. I proszę — ścisk jest taki sam jak przed dwoma laty. To świadczy o społecznym zainteresowaniu, a więc o potencjalnym popycie.

— Panie Juon, odpowiada pan za marketing i sprzedaż wyrobów „Stara" bodajże na jednej piątej powierzchni Ziemi. Jak zorganizowana jest pana praca?

— Jestem Szwajcarem i w zasadzie mieszkam w Szwajcarii. Mówię w zasadzie, ponieważ poza weekendami rzadko bywam w domu. Moje biuro znajduje się w RFN, ale i w nim nie spędzam zbyt wiele czasu. Odwiedzam np. wszystkie znaczące targi i wystawy, by znać na bieżąco tendencje na poszczególnych rynkach. Cały nasz zespół pracuje w wielu krajach i ma także wielonarodowy skład. Pan Musiał jest Polakiem z pochodzenia, wielu innych współpracowników to Francuzi, Niemcy. Przykład nasz daje najlepszą odpowiedź na pana pierwsze pytanie. Jesteśmy firmą bez granic.

Rozmawiał:
Grzegorz Onichimowski

DOKTOR NARESZCIE Z DYPLOMEM?



Firma Digital Research słynie na całym świecie głównie z opracowania i dystrybucji systemu CP/M. O tym, że dobra sława firmy nie musi świadczyć o niczym, przekonali się użytkownicy 8-bitowych Amstradów, zapoznawszy się z interpreterem DR Logo produkcji właśnie Digital Research. DR Logo w tej wersji to oprogramowanie niedostatecznie sprawdzone (niektóre procedury działają w ogóle błędnie), w którym powyrzucano z języka Logo, co tylko było można.

Honoru firmy stara się bronić DR Logo dla IBM PC. Choc nazwa ta sama, to jednak produkt jest zupełnie inny pod względem użytkowym. W niniejszym artykule opiszemy, co zyskali w postaci DR Logo użytkownicy IBM PC — a co stracili posiadacze Amstradow.

SPRZĘT

DR Logo dostarczane jest przez producenta na jednej, zabezpieczonej przed kopiowaniem dyskietce. Wymagana konfiguracja sprzętu to komputer z jedną stacją dysków i kartą graficzną CGA. W przypadku braku tej karty (gdy np. dysponujemy kartą Hercules) można nadal używać DR Logo, ale bez grafiki.

Uruchomienie programu odbywa się przez włożenie dyskietki Logo do stacji A., a następnie włączenie komputera lub wciśnięcie CTRL-ALT-DEL, jeżeli komputer był uprzednio włączony. Dyskietka Logo zawiera system operacyjny, zatem nie potrzebujemy wykonywać zadnych dodatkowych czynności.

Ale tu następuje przykra niespodzianka: tym systemem nie jest PC-DOS, lecz CP/M-86! Oznacza to, że nie możemy przenosić danych pomiędzy dyskietkami nagranymi pod systemem PC-DOS a dyskietkami nagranymi przez nas w Logo. Co gorsza, jeśli posiadamy twardy dysk, to nie możemy zainstalować na nim Logo, dopoki nie zostanie on sformatowany pod system CP/M-86. No, a to jest bez sensu — o czym wie każdy posiadacz twardego dysku na IBM PC, gdyz oznacza rezygnację z posiadania na tym dysku mnóstwa uzytecznych programow pracujących w systemie PC-DOS.

Użycie przez firmę własnego systemu operacyjnego odbyło się zatem ze szkodą dla wygody uzytkownika.

OGÓLNE CECHY JĘZYKA

W DR Logo przyjęto kilka załozeń, niespotykanych w innych implementacjach Logo. Dwie najważniejsze to:

1. Nazwy procedur pierwotnych piszemy zawsze małymi literami; słowa **"TRUE** i **"FALSE** nie są procedurami i jako takie muszą byc pisane duzymi literami, o ile mają oznaczac prawdę i fałsz.

2. Wiersz Logo nie musi odpowiadać pojedynczej linii, zakończonej wciśnięciem **ENTER**. Jeżeli linię rozpoczniemy od jednego lub więcej znaków spacji, to będzie ona kontynuacją poprzedniej

KLAWIATURA

F1 lub **CTRL-C** — Wyjście z edytora

F2 lub **CTRL-G** — Przerwanie wykonywania programu w Logo lub opuszczenie edytora bez utrwalenia wprowadzonych w nim zmian.

F3 lub **CTRL-W** — Zatrzymanie wykonywania wydruku na ekranie. Wznowienie wydruku następuje poprzez wciśnięcie dowolnego klawisza.

F4 lub **CTRL-Z** — Wstrzymanie wykonywania programu w Logo i powrót do trybu bezpośredniego. Kontynuację wykonania programu uzyskujemy przez użycie procedury **co**.

F5 lub **CTRL-K** — Usunięcie ciągu znaków począwszy od kursora aż do końca linii. Usuwane znaki umieszczane są w specjalnym buforze, z którego mogą zostać odtworzone przez wciśnięcie **F6**.

F6 lub **CTRL-Y** — Wypisanie tekstu zawartego w buforze, umieszczonego tam w wyniku ostatniego wciśnięcia **ENTER** lub **P5**.

F7 lub **CTRL-L** — W obrębie edytora: przesunięcie zawartości ekranu tak, aby linia wskazywana kursorem znalazła się w jego środku. Poza edytorem, wyświetlanie pełnego ekranu graficznego.

F9 lub **CTRL-A** — Przesunięcie kursora do początku linii.

F10 lub **CTRL-E** — Przesunięcie kursora na koniec linii.

CTRL-S — Przejście do trybu ekranu podzielonego między grafikę i tekst.

CTRL-T — Przejście do trybu ekranu tekstowego.

Klawisze kursora lub **CTRL-B,F,N,P** — Poruszanie kursorem

PgUp, PgDn lub **ESC-V, CTRL-V** — W obrębie edytora: poprzednia lub następna strona tekstu.

Home, End lub ESC- , — W obrębie edytora przesunięcie kursora na początek lub koniec tekstu.

Ins lub **CTRL-0** — Utworzenie nowej linii w miejscu kursora.

Del lub **CTRL-D** — Usunięcie znaku wskazywanego przez kursor.

← lub **CTRL-H** — Usunięcie znaku poprzedzającego kursor.

Tab lub **CTRL-I** — Wstawienie trzech spacji w miejscu kursora

NOWE POJĘCIA PIERWOTNE

bury ‹nazwa pakietu lub lista nazw pakietów›
Ukrycie wymienionego pakietu procedur przed działaniem następujących procedur:
edall, edns, edps, erall, erns, erps, glist, poall, pons, pops, pots, pps, save.

buttonp ‹numer joysticka›
Operacja dająca wynik **TRUE** wtedy i tylko wtedy, gdy wciśnięty jest wyszczególniony przycisk joystick-a. Poszczególnym przyciskom przyporządkowano numery:
0 — przycisk 1 joysticka 1
1 — przycisk 2 joysticka 1
2 — przycisk 1 joysticka 2
3 — przycisk 2 joysticka 2

catch ‹nazwa› ‹lista instrukcji›
Deklaracja obsługi błędu lub sytuacji wyjątkowej. Użyte w powiązaniu z **throw** pozwala na własnoręczne zorganizowanie obsługi takich przypadków. **Catch** deklaruje możliwość wystąpienia błędu o nazwie ‹nazwa›. W razie późniejszego wystąpienia w programie **throw** nazwa wykonywana jest podana przy **catch** lista instrukcji.

changef ‹nowa nazwa› ‹stara nazwa›
Zmiana nazwy pliku dyskowego. Oba argumenty są słowami

co
Zakończenie pauzy w programie (inaczej, wznowienie jest wykonywania od miejsca, w którym został przerwany poprzez wciśnięcie **CTRL-Z** lub napotkanie instrukcji **pause**).

copyd ‹stacja docelowa› ‹stacja źródłowa›
Wykonuje dokładną (ściezka po ścieżce) kopię dyskietki. Nie da się w ten sposób skopiować dyskietki z DR Logo!

copyf ‹nazwa kopii› ‹nazwa pliku›
Skopiowanie pliku

copyoff
Zatrzymuje proces kopiowania zawartości ekranu tekstowego na drukarce.

copyon
Rozpoczęcie kopiowania zawartości ekranu tekstowego na drukarce.

debug
Włącza tryb debuggera: ekran zostaje podzielony na dwie części. Dolna służy do wypisywania tekstów przez program Logo, górna obrazuje akcje wykonywane aktualnie przez interpreter.

defaultd
Operacja dająca w wyniku nazwę aktualnie używanej stacji dysków

degress ‹liczba radianów›
Operacja przeliczająca miarę kąta w radianach na miarę w stopniach.

edall

edall ‹nazwa pakietu lub lista nazw pakietów›
Przejście do edytora z załadowaniem wszystkich procedur (ew. wszystkich procedur wymienionych pakietów) znajdujących się aktualnie w pamięci.

erf ‹nazwa pliku›
Usunięcie z dysku pliku o podanej nazwie.

ern ‹nazwa zmiennej lub lista nazw zmiennych›
Usunięcie z pamięci wymienionych zmiennych

error
Operacja dająca w wyniku listę, określającą błąd, który ostatnio wystąpił. Zawiera ona sześć elementów:
1. Numer błędu.
2. Tekst komunikatu, odpowiadający temu numerowi.
3. Nazwę procedury, w której błąd wystąpił.
4. Tekst linii, w której błąd wystąpił.
5. Nazwę procedury występującą w błędnym wyrażeniu (o ile takowa występuje).
6. Treść parametru, w którym błąd wystąpił, o ile błąd wystąpił przy podaniu parametru.

exp ‹n›
Operacja, dająca w wyniku e^{n}.

fkey ‹n› ‹słowo›
Przypisanie klawiszowi funkcyjnemu o numerze ‹n› słowa podanego jako drugi parametr. Każde następne naciśnięcie klawisza F ‹n› spowoduje wypisanie przypisanego mu słowa. Słowo może zawierac znaki spacji, poprzedzone znakiem\, zgodnie z przyjętą w Logo konwencją.

follow ‹nazwa procedury› ‹nazwa procedury›
Przemieszczenie definicji procedur w pamięci tak, aby pierwsza z podanych procedur występowała przed drugą.

fullscreen
Powoduje przeznaczenie całego ekranu na wyświetla-

nie grafiki. Odpowiada wciśnięciu **CTRL-L**.
getfs
getfs ‹nazwa dysku›
Operacja, której wynikiem jest lista nazw plików na dysku. W przypadku braku parametru operacja dotyczy aktualnie używanej stacji dysków.
glist ‹nazwa własności›
glist ‹nazwa własności› ‹nazwa pakietu lub lista nazw pakietów›
Operacja, dająca w wyniku listę nazw wszystkich obiektów o własności podanej jako pierwszy parametr. Obiekty są wyszukiwane na obszarze wymienionych pakietów lub (gdy nie ma drugiego parametru) wśród wszystkich obiektów znajdujących się w pamięci.
go ‹słowo-etykieta›
Skok do miejsca programu, w którym znajduje się wywołanie **label** z takim samym parametrem. Skok nie może odbywać się w obrębie tego samego wiersza Logo — zatem nie wolno wykonywać skoku w obrębie pętli **repeat**, procedury **if**, **run** itp. Nie jest też dozwolony skok do innej procedury.
gprop ‹nazwa obiektu› ‹nazwa własności›
Operacja dająca w wyniku wartość podanej własności podanego obiektu. Jeśli dany obiekt nie ma podanej własności, to wynikiem jest lista pusta.
initd ‹nazwa dysku› ‹n›
Sformatowanie dysku znajdującego się w podanej stacji. Drugi parametr określa typ formatu:
1 — format jednostronny
2 — format dwustronny
keyp
Operacja dająca wynik TRUE wtedy i tylko wtedy, gdy aktualnie jest wciśnięty jakiś klawisz
label ‹słowo-etykieta›
Zdefiniowanie miejsca w procedurze, do której ma odbyć się skok przy pomocy **go**.
local ‹nazwa zmiennej›
Zdefiniowanie podanej zmiennej jako lokalnej, tzn. zażądanie usunięcia jej z pamięci po zakończeniu bieżącej instancji procedury. Procedura **local** jest zachłanna.
log ‹n›
Operacja logarytmu naturalnego.
log10 ‹n›
Operacja logarytmu o podstawie 10.
lowercase ‹słowo›
lc ‹słowo›
Operacja dająca w wyniku słowo wejściowe, w którym wszystkie duże litery alfabetu są zamienione na małe
lpen
Operacja dająca w wyniku listę współrzędnych pióra świetlnego na ekranie. Współrzędne odnoszą się do ekranu tekstowego, a nie do graficznego.
lpenp
Operacja wskazująca, czy pióro świetlne jest aktualnie w użyciu, tzn. czy można odczytać jego współrzędne.
nodebug
Wyłączenie trybu debuggera
noformat
Usunięcie komentarzy z tekstów wszystkich znajdujących się w pamięci procedur.
noprim
Usunięcie z pamięci informacji o procedurach pierwotnych. Po użyciu **noprim** wykonanie **poprim** nie daje żadnego efektu.
notrace ‹nazwa procedury lub lista nazw procedur›
notrace
Wyłącza śledzenie wszystkich lub wymienionych procedur.
nowatch
nowatch ‹nazwa procedury lub lista nazw procedur›
Wyłącza obserwowanie wszystkich lub wymienionych procedur.
package ‹nazwa pakietu› ‹nazwa obiektu lub lista nazw obiektów›
Utworzenie pakietu o podanej nazwie, zawierającego podane obiekty. Każda procedura i zmienna może należeć co najwyżej do jednego pakietu.
paddle ‹n›
Operacja dająca w wyniku liczbę określającą położenie manetki lub wiosełka. Parametr oznacza:
1 — współrzędna x wiosełka 1,
2 — współrzędna y wiosełka 1,
3 — współrzędna x wiosełka 2,
4 — współrzędna y wiosełka 2.
pause
Zatrzymanie wykonywania programu i przejście do trybu bezpośredniego, w celu np. umożliwienia np. pracy z edytorem. Aby zakończyć pauzę, należy wykonać procedurę **co**. Działanie **pause** odpowiada wciśnięciu **CTRL-Z**.
pen
Operacja dająca w wyniku listę dwuelementową, określającą stan pióra i jego kolor.
pi
Przybliżenie liczby PI, równe 3.14159265358979.
piece ‹a› ‹b› ‹obiekt›
Operacja, zwracająca listę lub słowo, powstałą przez „wycięcie" fragmentu listy lub słowa ‹obiekt›, poczynając od elementu o numerze ‹a›, a kończąc na ‹b›.
pkgall ‹nazwa pakietu›
Włącza wszystkie obecne w pamięcia procedury i zmienne, nie należące dotąd do innych pakietów, do pakietu o podanej nazwie.
plist ‹nazwa obiektu›
Operacja, dająca w wyniku listę własności posiadanych przez dany obiekt.
pocall ‹nazwa procedury›
Wyświetla nazwy wszystkich procedur wywoływanych przez podaną procedurę.
popkg
popkg ‹nazwa pakietu lub lista nazw pakietów›
Wyświetla nazwy i zawartości podanych (lub wszystkich) pakietów
poprim
Wyświetlanie nazw wszystkich procedur pierwotnych DR Logo. Wydruk jest przerywany co 24 linie w oczekiwaniu na wciśnięcie dowolnego klawisza.
poref ‹nazwa procedury lub lista nazw procedur›
Wyświetlenie nazw wszystkich procedur wywołujących podaną procedurę (procedury).
potl
Wyświetla nazwy procedur, które nie są wywoływane przez żadne inne procedury.
pprop ‹nazwa obiektu› ‹nazwa własności› ‹wartość własności›
Włączenie podanej własności wraz z jej wartością do listy własności danego obiektu.
pps ‹nazwa pakietu lub lista nazw pakietów›
Wyświetlenie wszystkich (poza systemowymi) własności i ich wartości wszystkich obiektów obecnych w pamięci.
printscreen
Skopiowanie zawartości ekranu graficznego na drukarkę. Wymagana jest drukarka o standardzie graficznym zgodnym z drukarkami Epson.
proclist
Operacja dająca w wyniku listę nazw wszystkich zdefiniowanych procedur.
quote ‹obiekt›
Operacja odpowiadająca użyciu znaku " w odniesieniu do ciągu znaków, z których składa się obiekt, np. **quote abcd** odpowiada **"abcd**.
radians ‹liczba stopni›
Operacja przeliczenia podanej miary stopniowej kąta na radiany.
readquote
rq
Operacja o działaniu podobnym do **readlist** — z tym, że wynikiem jest tu tekst wprowadzony z klawiatury „sklejony" w jedno słowo.
remainder ‹a› ‹b›
Reszta z dzielenia pierwszego parametru przez drugi.
remprop ‹nazwa obiektu› ‹nazwa własności›
Usunięcie podanej własności z listy własności danego obiektu. Uwaga: usunięcie obiektu z pamięci jest możliwe dopiero wtedy, gdy usunięto wszystkie własności z jego listy własności.
rerandom
Inicjalizacja generatora liczb losowych.
setd ‹nazwa dysku›
Ustalenie jako aktualnie używanego dysku o podanej nazwie
setpen ‹lista›
Ustalenie stanu pióra zgodnie z zawartością podanej listy. Pierwszy parametr (PENUP, PENDOWN, PENREVERSE lub PENERASE) określa stan pisaka, drugi (liczba całkowita) określa jego kolor.
setsplit ‹n›
Ustalenie ilości linii tekstu, dostępnych przy ekranie podzielonym między grafikę i tekst (**splitscreen**).
shuffle ‹lista›
Operacja, która daje w wyniku listę wejściową o losowo wymieszanych elementach.
sizef ‹nazwa pliku›
Operacja, której wynikiem jest długość danego pliku dyskowego w bajtach
spaced ‹nazwa dysku›
Operacja obliczająca ilość wolnych bajtów na podanym dysku
splitscreen
Podzielenie ekranu pomiędzy tekst i grafikę. Szerokość okna tekstowego można regulować przez **setsplit**.
textbg ‹n›
Ustalenie koloru tła ekranu tekstowego.
textfg ‹n›
Ustalenie koloru znaków na ekranie tekstowym.
throw ‹nazwa›
Powoduje wykonanie akcji związanej z podaną nazwą w uprzednio wywołanej procedurze **catch**.
tones ‹lista›
Zagranie dźwięku o parametrach zawartych w podanej liście. Składa się ona z dwóch liczb. Pierwsza określa częstotliwość dźwięku w Hz (koncertowemu A odpowiada liczba 440), druga — czas trwania dźwięku w milisekundach.
trace
trace ‹nazwa procedury lub lista nazw procedur›
Włączenie trybu śledzenia wymienionych (lub wszystkich) procedur. W tym trybie Logo komentuje na bieżąco wykonywane przez siebie czynności.
turtletext ‹słowo lub lista›
Wypisanie danego tekstu na ekran graficzny, począwszy od bieżącej pozycji żółwia.
twoscreen
Jeżeli komputer jest wyposażony w dwa monitory (kolorowy i monochromatyczny), **twoscreen** przeznacza monitor kolorowy tylko na grafikę, zaś monochromatyczny — tylko na tekst.
unbury ‹nazwa pakietu lub lista nazw pakietów›
Przywrócenie wymienionych pakietów działaniu procedur edycji, nagrania na dysk, wydruku. Zob. **bury**.
uppercase ‹słowo›
uc ‹słowo›
Operacja, której wynikiem jest słowo wejściowe, w którym wszystkie małe litery alfabetu zostały zmienione na duże.
watch
watch ‹nazwa procedury lub lista nazw procedur›
Włączenie trybu obserwacji w odniesieniu do wymienionych (lub wszystkich) procedur. W trybie tym Logo wyświetla na bieżąco treść interpretowanych wierszy. W odróżnieniu od **trace** nie są jednak opisywane czynności, wykonywane przez interpreter ani wartości zmiennych.
where
Operacja, której wynikiem jest numer elementu, odnalezionego w ostatnio użytej procedurze **memberp**.
‹a› ↑ ‹b›
Operacja potęgowania, używana w zapisie infiksowym, podobnie jak +, –, /, *
; ‹dowolny tekst›
Znak początku komentarza. Wszystkie znaki, począwszy od ; aż do końca linii nie są przez Logo interpretowane (oczywiście nie dotyczy to znaku średnika występującego np. wewnątrz list lub słów).

STARE POJĘCIA PIERWOTNE

Pod tym terminem rozumieć należy części języka Logo, które są wspólne dla niemal wszystkich dialektów Logo i których opis łatwo jest znaleźć w pierwszym lepszym podręczniku. Kto nie ma podręcznika, może odwołać się „Słownika minimum Logo" publikowanego w „Bajtku", w numerach 3—4, 5—6, 7 i 8/86, konfrontując go z listą procedur pierwotnych Dr. Logo, jaką możemy uzyskać przy pomocy **poprim**.

PODSUMOWANIE

DR Logo dla IBM PC wyraźnie przewyższa swoich „braci", działających w systemach CP/M, zarówno pod względem opracowania zbioru procedur pierwotnych, jak też jakości interpretera. W rozwoju Logo nie jest to jednak wcale szczyt możliwości. DR Logo prezentuje też kilka dotkliwych wad. Pierwszą jest konieczność używania go w systemie CP/M-86, co uniemożliwia wygodne zainstalowanie języka na twardym dysku. Niezbyt wygodny jest też edytor Logo — brakuje zwłaszcza możliwości operowania na blokach tekstu. Brak też możliwości dowolnego formatowania tekstu programu i jego komentowania. Dodanie do języka komentarzy „po średniku" nie rozwiązuje problemu.

Pomimo tego, że DR Logo nie daje się w roku 1988 zaliczyć do programów w pełni nowoczesnych, to jednak stanowi niewątpliwy skok jakościowy w stosunku do interpreterów Logo znanych z maszyn 8-bitowych. I jako takie warte jest poznania, zwłaszcza, że komputery „PC-podobne" coraz częściej trafiają do szkół, zaś rola Logo w edukacji została przez ostatnie dwa lata ugruntowana.

LITERATURA

1. **Dyskietka systemowa DR Logo** zawiera opis wszystkich procedur pierwotnych. Aby go przeczytać, należy po uruchomieniu Logo napisać **help**.
2. **Meet Dr. Logo for the IBM Personal Computer**, wyd. Digital Research, 1983. Początkowy kurs języka Logo.
3. **Dr. Logo Language Reference Manual for the IBM PC**, wyd. Digital Research, 1983. Oficjalny podręcznik Dr. Logo z dużą ilością drobnych błędów i niedociągnięć.
4. **LOGO — słownik minimum**, „Bajtek" nr 3/4—8 z r. 1986. Opis procedur pierwotnych pominiętych w niniejszym artykule.
5. Stanisław Waligórski **LOGO dla Sinclair Spectrum**, cz. I i II, wyd. IWZZ, 1987. Zwięzły wykład technik programowania w Logo, w większości aktualny także dla Dr. Logo.

Marek Wyrwidąb

XIO

Dołączona do komputera instrukcja Atari Basic wspomina o instrukcji XIO zaledwie tyle, że jest używana przy operacjach dyskowych i w grafice. Z tak lakonicznej informacji trudno cokolwiek zrozumieć, przybliżę więc Czytelnikom znaczenie tej ciekawej i pożytecznej instrukcji.

Format instrukcji

XIO jest skrótem nazwy „eXtended Input/Output command" — rozszerzona instrukcja wejścia/wyjścia. To już wiele wyjaśnia. Dokładniejszy opis zaczniemy jednak od formatu tej instrukcji. Ma ona postać:

XIO<rozkaz>,#<kanal>,<pomocn_1>, <pomocn_2>,"<urządzenie>"

gdzie

<rozkaz> — kod rozkazu wejścia/wyjścia;
<kanał> — numer kanału IOCB;
<pomocn_1> — pierwsza wartość pomocnicza;
<pomocn_2> — druga wartość pomocnicza;
<urządzenie> — nazwa urządzenia (dla operacji dyskowych nazwa pliku)

Zależnie od podanego kodu rozkazu instrukcja XIO może wykonywać różne operacje wejścia i wyjścia. Kolejno omówimy więc operacje ogólne, a następnie operacje XIO dla poszczególnych urządzeń.

Operacje ogólne

Wszystkie ogólne operacje XIO mają swoje odpowiedniki w innych instrukcjach Basica. Ich znaczenie jest następujące.

Instrukcja XIO	Instrukcja Basica	Operacja
XIO 3	OPEN	otwarcie kanału IOCB
XIO 5	INPUT	odczyt linii znaków
XIO 7	GET	odczyt jednego znaku
XIO 9	PRINT	zapis linii znaków
XIO 11	PUT	zapis jednego znaku
XIO 12	CLOSE	zamknięcie kanału IOCB
XIO 13	STATUS	odczyt statusu IOCB

Oprócz XIO 3 we wszystkich pozostałych obie wartości pomocnicze są ignorowane i równe zeru. W instrukcji XIO 3 (OPEN) pierwsza wartość pomocnicza określa rodzaj dostępu do urządzenia, a druga dodatkowe parametry — wielkość przesuwu w zapisie kasetowym dla "C:" (0 lub 128) i numer trybu graficznego dla "S:" (od 0 do 8). Rodzaj dostępu w zależności od urządzenia jest podany poniżej.

"C:" 4 — odczyt
8 — zapis

"D:" 4 — odczyt
6 — odczyt directory
8 — zapis nowego pliku
9 — zapis na końcu pliku
12 — równoczesny zapis i odczyt

"E:" 8 — zapis na ekranie
12 — zapis na ekranie i odczyt z klawiatury
13 — zapis i odczyt z ekranu

"K:" 4 — odczyt

"P:" 8 — zapis

"R:" 5 — odczyt równoległy
8 — zapis bloku
9 — zapis równoległy
13 — zapis i odczyt równoległy

"S:" podana wartość może być sumą poniższych liczb (obowiązkowa jest liczba 8, użycie pozostałych zależy od decyzji użytkownika):
4 — odczyt z ekranu
8 — zapis na ekranie
16 — utworzenie okna tekstowego (oprócz trybu 0)
32 — zawartość ekranu bez zmian (oprócz trybu 0)

Ponieważ istnieją odpowiednie instrukcje Basica o znacznie prostszej składni, rzadko używa się instrukcji XIO do wykonywania operacji ogólnych. Jest to więc swego rodzaju ciekawostka Atari Basic.

Operacje graficzne

Dwie operacje XIO mogą być wykonywane wyłącznie na ekranie: rysowanie linii (XIO 17 = DRAWTO) i wypełnianie obszaru (XIO 18). Pierwszej z nich odpowiada instrukcja Basica DRAWTO i nie będzie ona nas interesować. Druga, nie mająca odpowiednika w Basicu, udostępnia dodatkową interesującą funkcję graficzną. Ponieważ jest to najczęściej stosowana instrukcja XIO, to jej format podam osobno:

XIO 18,#6,0,0,"S:"

Użycie jej nie jest proste, więc zademonstrowane zostanie na przykładzie. Najpierw należy narysować linię ograniczającą z prawej strony wypełniany obszar i umieścić kursor instrukcją PLOT lub DRAWTO w miejscu, z którego rozpoczniemy wypełnianie (linie 90 i 100). Następnie do rejestru FILDAT (765) wpisujemy numer koloru, który ma zostać użyty do wypełniania i umieszczamy kursor instrukcją POSITION w miejscu zakończenia wypełniania (linia 110). Dopiero teraz można wywołać instrukcję XIO 18 (linia 120). W liniach 20—80 ustalany jest kolor i losowo wybierane jest miejsce rysowania. Cała procedura jest powtarzana po naciśnięciu klawisza START.

Należy pamiętać, że XIO 18 wypełnia każdą linię obrazu, aż do napotkania punktu o kolorze różnym od koloru tła. Nie można więc wypełniać obszarów wypełnionych już wcześniej.

Operacje dyskowe

Trzecią grupą operacji, które można wykonać przy użyciu instrukcji XIO, są operacje dyskowe. Normalnie konieczne jest wczytanie DOS-u (plik DUP.SYS) i wybranie tych funkcji z menu DOS-u. Zastosowanie instrukcji XIO pozwala na osiągnięcie tych funkcji z poziomu Basica. We wszystkich operacjach wartości pomocnicze są ignorowane i powinny być równe zeru. Oto pełny wykaz operacji dyskowych XIO:

Instrukcja XIO	Instrukcja DOS-u	Operacja
XIO 32	RENAME	zmiana nazwy pliku
XIO 33	DELETE	skasowanie pliku
XIO 35	LOCK	zabezpieczenie pliku
XIO 36	UNLOCK	odbezpieczenie pliku
XIO 37	POINT (Basic)	ustawienie głowicy
XIO 38	NOTE (Basic)	odczyt ustawienia głowicy
XIO 254	FORMAT	formatowanie dyskietki
XIO 253	FORMAT SINGLE	formatowanie dyskietki w pojedyńczej gęstości

Operacje XIO 37 i XIO 38 odpowiadają instrukcjom Basica POINT i NOTE i wymagają uprzedniego otwarcia pliku instrukcją OPEN (lub XIO 3).

Podane wyżej operacje nie wyczerpują możliwości instrukcji XIO. Istnieje jeszcze jedna grupa operacji, które mogą być przez nią wykonywane. Są to operacje przesyłania danych przez interfejs szeregowy RS232. Ponieważ interfejsy takie są w Polsce prawie niespotykane, to ta grupa operacji XIO została pominięta. Zainteresowanych odsyłam do książki Wiesława Miguta „Atari Basic".

Wojciech Zientara

```
NM  10 GRAPHICS 31
KH  20 X1=INT(RND(0)*120)+40
OB  30 Y1=INT(RND(0)*162)+30
PI  40 X2=INT(RND(0)*140)
XE  50 IF X2>=X1 OR X2<X1-60 THEN 40
SM  60 Y2=INT(RND(0)*180)
BT  70 IF Y2>=Y1 OR Y2<Y1-60 THEN 60
DE  80 C=C+1:IF C>3 THEN C=1
FU  90 COLOR C:PLOT X1,Y1
LW  100 DRAWTO X1,Y2:DRAWTO X2,Y2
BS  110 POKE 765,C:POSITION X2,Y1
RZ  120 XIO 18,#6,0,0,"S:"
FU  130 ON (PEEK(53279)=6)+1 GOTO 130,20
```

PRZESUWANIE KURSORA

Program, który chciałbym zaproponować powstał w wyniku wykorzystania wiadomości zawartych w cyklu pt. „Nie bój się przerwań".

Służy on do przesuwania kursora przy pomocy joysticka, a także umożliwia usuwanie z ekranu tekstu, który znajduje się po prawej stronie kursora. Naciśnięcie przycisku FIRE spełnia funkcję CTRL-DELETE. Procedura w języku maszynowym wykonywana jest co trzy przerwania VBLK. Sprawdza ona położenie joysticka i stosownie do niego umieszcza w komórce 764 wartość kodu wewnętrznego oznaczającego wciśnięcie CTRL i odpowiedniego klawisza przesuwania kursora. Gdy nie poruszamy joystickiem wszystkie funkcje klawiatury działają normalnie.

Procedura ta może mieć zastosowanie np. podczas pisania, sprawdzania i poprawiania listingów programów w języku Basic, ponieważ pozwala na szybkie umieszczenie kursora w dowolnym miejscu ekranu i wpisanie tam odpowiednich znaków. Po dokonaniu korekty naciśnięcie RETURN umieszcza w pamięci poprawioną linię.

Wciśnięcie RESET przerywa działanie procedury. Ponowne jej uruchomienie następuje przez podanie instrukcji X = USR(1619)

Andrzej Holanowski

```
NJ  10 READ A:IF A=-1 THEN 40
LZ  20 POKE 1536+I,A:I=I+1
RK  30 GOTO 10
EU  40 POKE 54286,64:POKE 206,3
ZD  50 I=USR(1619):NEW
MA  100 DATA 198,206,165,206,208
SU  110 DATA 74,173,120,2,201,15
JB  120 DATA 208,15,169,255,141
ZK  130 DATA 252,2,173,132,2,208
BS  140 DATA 5,169,180,141,252,2
WD  150 DATA 173,120,2,201,13,208
BI  160 DATA 5,169,143,141,252,2
XG  170 DATA 173,120,2,201,14,208
AU  180 DATA 5,169,142,141,252,2
XI  190 DATA 173,120,2,201,7,208
BQ  200 DATA 5,169,135,141,252,2
TY  210 DATA 173,120,2,201,11,208
BC  220 DATA 5,169,134,141,252,2
TY  230 DATA 169,3,133,206,76,98
XQ  240 DATA 228,104,160,0,162,6
GY  250 DATA 169,7,76,92,228,-1
```

GRAFIKA W DLI

Atari posiada bardzo duże możliwości graficzne. Niestety z poziomu BASIC-a ilość kolorów dostępna jednocześnie na ekranie wynosi w najlepszym przypadku 16.

Przedstawiony w artykule program wykorzystujący przerwania niemaskowalne DLI generowane przez ANTIC umożliwia zmianę zawartości rejestrów koloru w poszczególnych liniach. Daje to maksymalną ilość 256 kolorów na ekranie.

Przerwania DLI są wywoływane przy czytaniu programu ANTIC-a. Bliższe informacje dotyczące programowania tego układu zawarte są w „Komputerze" 8/86. Przerwania są wywoływane przez dowolnym rozkazie ANTIC-a, który ma ustawiony najstarszy, siódmy bit. Adres programu obsługującego przerwania wskazywany jest przez wektor DLIVKT (512, 513).

Wróćmy do programu. Niżej przestawione są cztery wydruki. Po przepisaniu pierwszego i uruchomieniu otrzymujemy program, który przedstawia wydruk drugi. Są to dwa programy w języku wewnętrznym. Pierwszy modyfikuje i lokuje program przerwań w odpowiednim miejscu oraz ustawia wektor. Jest on wywoływany przez:

A = USR (ADR (OBS) + 52, ADR (OBS), ADPR, ILO, ADDA)

gdzie:

ADPR — adres, od którego ma być umieszczony program obsługi przerwań DLI — program o długości 52 bajtów;

ILO — ilość uaktywnionych przerwań;

ADDA — obszar początku danych dla programu.

Pierwszy bajt obszaru danych zajmuje licznik aktualnie obsługiwanego przerwania. Reszta obszaru jest podzielona w następujący sposób:

Adres	Numer obsługiwanego przerwania	Zapisywany rejestr
ADDA + 1	1	COLPF0 (708)
ADDA + 2	2	COLPF0 (708)
......	......	......
ADDA + 1 + ILO	1	COLPF1 (709)
......	......	......
ADDA + 1 + ILO 2	1	COLPF2 (710)
......	......	......
ADDA + 1 + ILO 3	1	COLBAK (712)

Drugi program w języku wewnętrznym jest wywoływany przez:

POKE ADDA,0:A = USR(ADR(OBS) + 156,ADR(OBS) + 171)

i uaktywnia przerwania DLI w momencie przerwania synchronizacji pionowej.

Wydruk trzeci to procedura zmieniająca program ANTIC-a i uruchamiająca przerwania. Działa w dowolnej grafice. Przed uruchomieniem jej w pamięci musi się znajdować program, który przedstawia drugi wydruk. Jedynki w ciągu C1$ reprezentują linie na ekranie, w których mają wystąpić przerwania, zera — linie, w których nie będzie przerwań. Przed uruchomieniem procedury trzeba umieścić w pamięci, poczynając od ADDA + 1 kolory poszczególnych linii. W przeciwnym razie większość ekranu po uruchomieniu procedury zmieni kolor na czarny.

UWAGI:

1. Program obsługi przerwań DLI jest wywoływany zawsze po zakończeniu rysowania przez ANTIC linii na ekranie i np. przy włączonym przerwaniu w pierwszej linii zmiana koloru nastąpi w drugiej.
2. Zmiana koloru w jednej linii powoduje zmianę koloru także w następnych, w których nie ma uaktywnionych przerwań.
3. Przed wywołaniem drugiego programu w języku wewnętrznym program ANTIC-a powinien być już zmieniony.
4. Chcąc wyłączyć przerwania trzeba albo zmienić tryb graficzny, albo wpisać instrukcję POKE 54286,127, albo nacisnąć RESET.

Czwarty wydruk jest krótkim programikiem demonstrującym możliwości, jakie daje zastosowanie opisywanych przerwań DLI.

Wojciech Wylon

```
TI  10 DIM A$(182):FOR A=0 TO 6:S=0:FOR V=
    1 TO 26:READ B:A$(A*26+V)=CHR$(B):S=S+
    B:NEXT V
RR  20 READ C:IF C<>S THEN ? "BLAD W LINII
    ";(9+A)*10:LIST (9+A)*10:END
YH  30 NEXT A:? CHR$(125):POSITION 2,7:? "
    5000 DIM OBS$(182)"
DM  40 ? "5001 OBS$=";CHR$(34);A$(1,93);CHR
    $(34):? "5002 OBS$(94)=":
PQ  50 ? CHR$(34);A$(94);CHR$(34):POSITION
    2,22:? "CONT":POSITION 0,0:POKE 842,1
    3:STOP
HN  60 POKE 842,12:ILO=2:ADDA=30000
SJ  70 ? CHR$(125):POSITION 2,23:? "POKE 8
    42,12:LIST";:POSITION 2,4
DV  80 FOR A=10 TO 150 STEP 10:? A:NEXT A:
    POSITION 0,0:POKE 842,13:STOP
PD  90 DATA 169,0,141,0,0,240,4,8,72,138,7
    2,173,0,0,201,0,240,238,234,170,189,0,
    0,141,22,208,2660
UB  100 DATA 189,0,0,141,23,208,189,0,0,14
    1,24,208,189,0,0,141,26,208,238,48,117
    ,104,170,104,40,64,2572
XS  110 DATA 104,104,133,204,104,133,203,1
    04,133,206,141,1,2,104,133,205,24,105,
    7,141,0,2,144,3,238,1,2579
LZ  120 DATA 2,160,0,177,203,145,205,200,1
    92,52,208,247,104,104,133,207,160,15,1
    45,205,104,160,4,145,205,160,3542
HD  130 DATA 13,145,205,160,46,145,205,170
    ,104,136,145,205,160,3,145,205,160,12,
    145,205,24,105,1,144,1,232,3221
ZM  140 DATA 160,21,145,205,72,200,138,145
    ,205,104,24,101,207,144,1,232,200,200,
    200,200,200,192,45,208,233,96,3878
YA  150 DATA 169,1,141,24,2,104,104,141,39
    ,2,104,141,38,2,96,169,192,141,14,212,
    95,2,0,8,0,0,1932
```

Listing 1

```
UD  5000 DIM OBS$(182)
SM  5001 OBS$="[illegible]"
KI  5002 OBS$(94)="[illegible]"
```

Listing 2

```
EA  8000 DIM C1$(100),C2$(16)
VA  8001 C2$="25789:;=????45<>"
LP  8002 C1$="0111111111111111100009010"
TJ  8010 DL=PEEK(560)+256*PEEK(561)-1:P0=A
    SC(C2$(PEEK(87)+1))-48:P1=0:P2=0:P3=0
WY  8011 P4=PEEK(DL+P1):IF P4=65 OR P2=LEN
    (C1$) THEN 8014
GE  8012 IF P4=P0 OR P4=P0+64 THEN P2=P2+1
    :IF VAL(C1$(P2,P2)) THEN POKE DL+P1,P4
    +128:P3=P3+1
OM  8013 P1=P1+1:GOTO 8011
PH  8014 ADPR=1536:ADDA=30000:PRZE=P3:POKE
    ADDA,0
UQ  8015 V=ADR(OBS$):A=USR(V+52,V,ADPR,PRZE
    ,ADDA):A=USR(V+156,V+171)
```

Listing 3

```
JK  5 GRAPHICS 18
AM  7 POKE 708,240:POKE 709,255:POKE 710,2
    44:POKE 711,250:POKE 712,242
LP  10 POSITION 4,3:? #6:"DEMO":? #6;"
    256 KOLOROW":? #6;"              [illegible]":? #6
    ;"          atari"
LI  17 FOR A=0 TO 16:POKE 30052+A,A*16-256
    *(A=16):NEXT A
SO  20 OPEN #1,4,0,"K:":GET #1,V
GN  25 GRAPHICS 9
HJ  30 FOR A=-15 TO 15:COLOR 15-ABS(A):FOR
    B=0 TO 1:PLOT 40+A*2+B,0:DRAWTO 40+A*
    2+B,190:NEXT B:NEXT A
PO  35 POKE 53277,1:POKE 53261,255:POKE 53
    262,255:POKE 53248,36:POKE 53249,191
ZB  37 POKE 53256,255:POKE 53257,255:POKE
    708,0:POKE 709,0
KO  40 DL=PEEK(560)+256*PEEK(561):FOR A=10
    TO 190 STEP 12:POKE DL+A,128+15:NEXT
    A
VV  50 POKE DL+197,128+15:POKE DL+2,140:PO
    KE DL-1,112:POKE 560,PEEK(560)-1
UD  5000 DIM OBS$(182)
SM  5001 OBS$="[illegible]"
KI  5002 OBS$(94)="[illegible]"
IA  6004 ADPR=1536:ADDA=30000:PRZE=17:POKE
    ADDA,0
UL  6005 V=ADR(OBS$):A=USR(V+52,V,ADPR,PRZE
    ,ADDA):A=USR(V+156,V+171)
OH  7000 FOR A=0 TO 255:POKE 7[illegible]
    4,A:POKE 30260,A:POKE 705,A
WV  8000 FOR G=0 TO 10:NEXT G:NEXT A:GOTO
    7000
```

Program demonstracyjny

OBSŁUGA KLAWISZA RESET

RESET jest przyciskiem zerującym procesor, co nie oznacza jednak, że nie można jego obsługi programować z poziomu kodu maszynowego.

Procedura taka, w odróżnieniu od procedury wywoływanej rozkazem USR, nie może zawierać rozkazu PLA, a podobnie jak procedura USR, powinna się kończyć rozkazem RTS. Po napisaniu programu maszynowego i umieszczeniu go w pamięci trzeba zmienić zawartość komórek 12 (młodszy bajt) i 13 (starszy bajt adresu procedury) oraz wpisać do komórki 9 wartość zezwalającą na wykonanie procedury (POKE 9,1). Radziłbym jednak procedurę taką wypróbować instrukcją USR, gdyż błąd w niej spowoduje zawieszenie się komputera i utratę danych.

Przykładem wykorzystania tych informacji jest poniższy program, który powoduje, że naciśnięcie klawisza RESET nie przerywa działania programu, a jest jednoznaczne z rozkazem RUN.

Litery przed numerem linii to sumy kontrolne Edytora BASICa, nie należą one do programu i nie należy ich wpisywać.

Program ten należy zapisać na taśmie rozkazem LIST „C", co umożliwi dołączenie tej procedury do innych programów rozkazem ENTER "C":

Jakub Cebula

```
AU  1 FOR I=1536 TO 1590:READ A:POKE I,A:N
    EXT I
CI  2 POKE 206,PEEK(88):POKE 207,PEEK(89)+
    1:POKE 9,1:POKE 12,1:POKE 13,6
PR  3 DATA 104,162,0,142,198,2,162,0,142,1
    97,2,160,26,185,8,6,145,206,200,192,46
    ,208,246,162,13
VY  4 DATA 142,74,3,96,0,0,0,0,0,48,47,43,
    37,0,24,20,18,12,17,18,2[illegible],[illegible],50,14,16,
    26,50,53,46,0
```

NIE TYLKO DLA GRACZY

Przedstawiony program składa się z sześciu niezależnych części ze względów praktycznych połączonych razem. Może on być pomocny użytkownikom ATARI 800 XL (65 XE), ktorzy zajmują się programowaniem w kodzie maszynowym, dostosowywaniem innych programów do swoich potrzeb, „rozgryzaniem" gier itp.

Po załadowaniu programu z taśmy (CLOAD, czas 1 minuta 50 sekund) i uruchomieniu (RUN) zgłasza się opcja ładowania innego programu z taśmy (krótka przerwa między blokadami). Długość programu można wstępnie ocenić wg. wzoru:

$$B = \frac{T-20}{20}$$

gdzie: T — całkowity czas nagrania wczytywanego programu w sekundach,
B — szukana długość programu w KB.

Jest to o tyle istotne, że zbyt długi program wczyta się do pamięci ekranu a dalsza część nie zostanie w ogóle zapisana, gdyż za obrazem jest ROM.

Po zakończeniu wgrywania programu wyświetla się liczba wczytanych bajtów. Program prezentuje swoje możliwości po wciśnięciu klawisza ESC; naciśnięcie spacji jest rozumiane jako rezygnacja z „instrukcji obsługi". Do opisu podprogramów można wrócić naciskając HELP **podczas** realizacji jednego z nich.

Następne opcje to:

po G. 100 „rozszyfrowywanie" zajmowania pamięci w komputerze przez program dyskowy wczytywany z kasety,
po G. 400 wyświetlenie zawartości pamięci.
po G. 500 zapis do pamięci,
po G. 700 zapis na taśmę wybranego obszaru pamięci,
po G. 800 przemieszczanie wybranego obszaru pamięci.

Programu można użyć jako kopiera.

Nie ma tu żadnych ukrytych tricków i można wprowadzać dowolne zmiany. Program zajmie trochę mniej pamięci, jeżeli przepisze się go łącząc niektóre linie w jedną, używając skrótów oraz pisząc na ekranie bez marginesów (POKE 82,0); można też opuścić wszystkie komentarze.

Wszystkim zainteresowanym życzę cierpliwości przy przepisywaniu i sukcesów w dalszych pracach.

Marek Renner

```
AY  1 REM  ZAPIS Z [illegible]. Podajac adres
           pamietaj,ze ten program zajmuje
           pamiec od [illegible] oraz ze
UK  2 REM  [illegible] zapisza sie dane siegajace
           poza adres 39967.
YO  3 ? "K":POSITION 2,6:? " Podaj adres p
    oczatku zapisu z tasmy ":CLR :POSITIO
    N 17,7:INPUT POCZ:DIM A$(6):A$="h[illegible]LV[illegible]
    "
ZP  4 POSITION 2,6:? "[illegible] wpro
    wadzony od adresu                 "
CI  5 POSITION 23,7:? "
                                      "
TN  6 POCZS=INT(POCZ/256)
DL  7 POCZM=POCZ-POCZS*256
MR  15 OPEN #1,4,120,"C:":POKE 850,7
FE  20 POKE 856,0:POKE 857,PEEK(742)-PEEK(
    745):POKE 853,POCZS:POKE 852,POCZM
UF  25 C=USR(ADR(A$))
ZY  30 ? "WCZYTALO SIE ";PEEK(856)+256*PEE
    K(857);" BAJTOW   "
HX  32 IF PEEK(851)<>136 AND PEEK(851)<>1
    THEN ? "!!!  BLAD nr. ";PEEK(851)
LT  35 CLOSE #1
FL  39 ? " Dalsze opcje programu [illegible]
    jeciu               klawisza <(ESC)>
           "
JD  40 DATA 0,2,100,119,400,402,500,504,70
    0,709,800,809
JZ  42 RESTORE 40
TU  44 FOR IK=0 TO 5
SA  46 READ OD,AZ
VM  48 GOSUB 62
MY  50 ? "K":LIST OD,AZ
KI  52 FOR I=0 TO 20:POSITION 0,I:? "
       ":NEXT I:POSITION 12,23:? "<ESC> l
    ub <SPACJA>"
PF  54 POKE 764,255
VJ  56 GOSUB 62
MF  58 NEXT IK
LF  60 ? "K":? :? :? :? "         TO WSZYSTKO
    - TERAZ PRACUJ         ":GOSUB 900:END
LZ  62 FOR I=0 TO 200
DO  64 IF PEEK(764)=33 THEN POP :? "K":END
YP  66 IF PEEK(764)<>28 THEN FOR II=0 TO 1
    00:NEXT II:NEXT I:? "K":END
AM  68 RETURN
YY  100 REM [illegible] w liniach 120-[illegible] od-
            czytuje w jakich adresach be-
            dzie zapisany czytany program.
XI  105 REM Trzeba jednak miec pewnosc, [illegible]
            w [illegible] programu nie sa ustalone [illegible] adresy
WQ  110 REM [illegible] pierwszego
            z czterech bajtow adresowych,
            adres poczatku i konca zapisu
PV  115 REM danego bloku oraz liczbe baj-
            tow do zapisu.Nr. [illegible] bajt za-
            pisany w adresie podanym przez
VD  117 REM Ciebie po [illegible] tego
            programu.
            [illegible] po G.[illegible]
HI  120 POKE 82,0
UH  125 I=POCZ:? PEEK(I),PEEK(I+1):I=I+2
XR  130 A=PEEK(I):B=PEEK(I+1):C=PEEK(I+2)
CT  135 D=PEEK(I+3)
HV  140 E=256*B+A:REM adr.pierwszego bajtu
OG  150 F=256*D+C:REM adr.ostatniego bajtu
WZ  160 G=F-E+1:REM liczba [illegible]tow do wczy-
                    tania.
ZK  170 ? "Nr.";I-POCZ+1;".";"←←←))←←←←←";
    E;" do ";F;"←←))←←←←←←= ";G;"←←←←←←←←←←
    ))←←←←←←bajtow"
TV  177 I=I+G+4:IF PEEK(I)<>0 THEN GOTO 19
    0
NE  178 IF PEEK(I+1)<>0 THEN GOTO 190
NW  179 IF PEEK(I+2)<>0 THEN GOTO 190
EM  180 IF PEEK(I+3)=0 THEN GOTO 200
NF  190 GOTO 130
BF  200 ? "))  ———————————————————)))
      RAZEM  ";I-POCZ+1;"←)  BAJTOW"
KR  210 POKE 82,2:GOSUB 900:END
WM  400 REM [illegible]
            [illegible]
            [illegible] pierwszy ma nr.1)
FD  401 REM ilosci bajtow do odczytania
            i adresu,od ktorym ma byt
            umieszczony ten bajt w pracu-
JC  402 REM jacym programie, zostanie wys-
            wietlona zawartosc obszaru
            pamieci.
OC  403 ? "Nr.bajtu,liczba bajtow,adres ";
KH  404 INPUT X,Y,ADRES
AA  405 ZZ=X-1:ZZZ=ZZ+POCZ-1
YV  410 FOR Z=1 TO Y STEP 20
YV  411 FOR YY=Z TO Z+19:? ZZ+YY;" > ";ADR
    ES+YY-1;" >>  ";PEEK(ZZZ+YY):NEXT YY
EE  412 IF PEEK(764)=255 THEN 412
XQ  413 POKE 764,255:? "K"
NO  420 NEXT Z:GOSUB 900:GOTO 400
RT  500 REM ZAPIS DO PAMIECI. (po G.5[illegible])
            Podaj adres poczatkowy.Dalsze
            liczby beda zapisywane od
LL  501 REM tego do nast[illegible].
BP  505 ? " ADRES ":INPUT A
GR  506 ? " TERAZ TYLKO DANE "
AL  510 INPUT B:POKE A,B:? ,,A:A=A+1
EY  520 GOSUB 900:GOTO 510
GX  700 REM ZAPIS NA KASETE.   (po G.700)
            Nalezy podac adres poczatkowy
            w [illegible]) oraz [illegible]
EL  705 REM [illegible]tow do wyslania (ILE!)
QR  710 CLR :CLOSE #1:DIM Z$(6):? "SKAD";:
    INPUT SK:? "ILE";:INPUT IL:SKS=INT(SK/
    256):SKM=SK-SKS*256:ILS=INT(IL/256)
ON  711 ILM=IL-ILS*256
TI  715 ? "SKM  ";SKM,"SKS  ";SKS:? "ILM
    ";ILM,"ILS  ";ILS
VK  720 Z$="h[illegible]LV[illegible]":OPEN #1,8,128,"C:":POK
    E 850,11:POKE 856,ILM:POKE 857,ILS:POK
    E 852,SKM:POKE 853,SKS
YL  730 I=USR(ADR(Z$)):CLOSE #1:GOSUB 900:
    END
LL  800 REM PRZEMIESZCZANIE ZAPISU (G.800)
            (wg. komunikatow)
LH  810 ? " Skad przepisywac       ";:INPUT
    B:? " Dokad przepisac       ";:INPUT I:
    ? " Ile bajtow przepisac ";:INPUT C
OP  820 POKE I,PEEK(B):I=I+1:B=B+1:C=C-1:I
    F C=0 THEN ? :? " Uff!! Gotowe! ":GOSU
    B 900:END
PH  830 GOTO 820
AA  900 IF PEEK(732)<>17 THEN RETURN
CY  910 POKE 732,0:GOTO 39
```

ATARI Basic

Duża popularnosć komputerów z rodziny Atari spowodowała ogromne zainteresowanie literaturą poświęconą tym komputerom. Krajowa Agencja Wydawnicza wydała właśnie książkę „Atari Basic" pod redakcją Wiesława Miguta.

Książka ta jest dobrze znana użytkownikom Atari, bowiem ukazała się już w 1986 roku w formie skryptu wydanego przez Ośrodek Doskonalenia Kadr Technicznych Rady Stołecznej NOT i przez prawie 2 lata pełniła rolę jednego z nielicznych źródeł wiedzy o tym komputerze. W nowej starannej szacie graficznej, po wprowadzeniu pewnych poprawek i uzupełnień prezentuje się znacznie lepiej pod względem estetycznym, a o wiele wyższa jakość druku wpłynęła korzystnie na przejrzystość informacji.

Autorzy starali się przedstawić możliwie wszechstronnie komputer, jego zastosowania, budowę i współpracujące z nim urządzenia. Książka zawiera wyczerpujący opis Basica ilustrowany przykładami, z uwzględnieniem specyficznych możliwosci generowania dźwięku i wykorzystania grafiki. Kilka zagadnień omówiono bardziej szczegółowo, a wśród nich animację obrazu, metody sortowania i niektóre procedury obliczeń matematycznych. Brakuje jednak rozdziału wyjaśniającego ogólne zasady budowy programu. Krótka wzmianka o stylu programowania nie wystarcza, by przygotować czytelnika do samodzielnego napisania dobrego, ciekawego programu.

Dla użytkownika duże znaczenie ma nie tylko znajomość języka programowania. Często o wiele ważniejsza jest umiejętność sprawnego posługiwania się zgromadzoną biblioteką oprogramowania. Z tego względu należy pochwalić zespół autorów za dołączenie szczegółowego omówienia dyskowego systemu operacyjnego.

W książce znajduje się również coś dla wielbicieli **peek**'ów i **poke**'ów: mapa pamięci i opis ważniejszych komórek. Niestety, informacje te na pewno nie zadowolą bardziej dociekliwego czytelnika. Autorzy zamieścili również listę rozkazów mikroprocesora 6502, bardzo pożyteczny materiał, ale mało przydatny dla użytkownika, który jest właśnie na etapie poznawania komputera i Basica.

Do książki można dokupić kasetę lub dyskietkę z programami. Pomoce te bardzo ułatwią życie niecierpliwym osobom, które nie lubią przepisywać wydruków programów.

W sumie pracę zespołu kierowanego przez Wiesława Miguta należy ocenić dość dobrze. Jest ona bardzo przydatna dla użytkowników Atari, szczególnie tych początkujących. Szkoda tylko, że — jak zwykle — książka ta ukazuje się z ogromnym opóźnieniem. Wcześniejsze wydanie uchroniłoby autorów przed bezprawnym sprzedaniem na giełdach setek drogich odbitek kserograficznych.

(j.j.)

„Atari Basic", praca zbiorowa pod redakcją Wiesława Miguta, KAW, Warszawa 1987, wyd. 1, nakład 50 000 egz. cena 550 zł.

AMIGA

SUPERKOMPUTER POD STRZECHY



Amiga 2000

Amiga 2000 jest systemem otwartym. Zamontowanie dodatkowych kart z procesorami INTEL 8088 lub 80286 przekształca Amigę w komputer kompatybilny, z IBM PC/XT lub AT — zależnie od karty. Komputer możemy rozbudowywać dalej. Mamy pełen wybór jeżeli chodzi o instalowanie dodatkowych stacji dysków: dwie stacje 3.5" lub 5 1/4", dysk twardy lub też inne przystawki i karty.

Z jednostką centralną Motorola 68000 współpracują trzy dodatkowe procesory zajmujące się grafiką i animacją, dźwiękiem oraz operacjami wejścia/wyjścia. Sam system jako niezwykle elastyczny i otwarty umożliwia dołączenie 9 dodatkowych kart (razem z już wymienionymi), na których może znajdować się rozszerzenie pamięci do 2 MB, dodatkowe koprocesory itp. Istnieje również możliwość wymiany procesora 68000 na 32-bitowy procesor Motorola 68020. Procesor 68881 i układ zarządzający pamięcią stają się niezbędne gdy myślimy o stosowaniu systemu UNIX.

Nowatorstwo Amigi 2000 polega na współpracy dwóch systemów: INTEL i MOTOROLA czyli MS-DOS i AMIGA-DOS lub jak kto woli IBM PC/XT/AT i AMIGA. Współpracę tę umożliwia bridgeboard czyli karta koordynująca współpracę obu systemów. Bridgeboard umożliwia zastosowanie tysięcy programów sprawdzonych na komputerach IBM.

Amiga 2000 umożliwia również tzw. multitasking czyli obsługę kilku programów jednocześnie. Powoduje to, że bez najmniejszych kłopotów możemy połączyć kilka programów i korzystać z nich jak gdyby to był jeden oryginany program.

Amiga 2000 to również fenomenalna grafika. Procesor MOTOROLA 68000 wspomagany przez 3 pozostałe koprocesory pozwala na tworzenie płynnie animowanej grafiki trójwymiarowej, definiowanie poruszających się obiektów (znane z komputerów domowych sprite'y). Maksymalna rozdzielczość wynosi 640 x 512 punktów (wykorzystujemy wtedy 16 kolorów z palety 4096 kolorów). Pozostałe tryby pracy procesora graficznego to 320 x 256 (32 kolory), 320 x 512 i 640 x 256. Jest więc z czego wybierać.

Dodatkowy interface „Genlock" umożliwia współpracę komputera z telewizorem lub wideo. Obraz z wideo możemy wzbogacać o elementy graficzne teksty wyjaśniające itp. wykorzystując wbudowane instrukcje graficzne.

Amiga 2000 posiada możliwość stereofonicznego odtwarzania dźwięków dowolnego instrumentu. Może również czytać zadany tekst głosem męskim, kobiecym lub „mechanicznym" — wyraźnym i dobrze modulowanym.

DANE TECHNICZNE

CPU: Motorola 68000, 16/32 bity (7,16 MHz)

3 koprocesory standardowe wbudowane do obsługi DMA, wejścia/wyjścia, video, grafiki i dźwięku.

Pamięć: IBM rozszerzalna do 9MB

Stacja dysków 3.5" (880 KB). Pozostawiono miejsce w komputerze dla dodatkowej stacji 3.5", dla dysku twardego lub dla dysku twardego i stacji 5 1/4".

Wyjścia: klawiatura; mysz, pioro świetlne; joystick; wiosełka; RS-232; programowalne złącze rownoległe (konfiguracja Centronics); wyjścia stereofoniczne (CINCH) oraz wideo RGB (analogowe lub cyfrowe); zewnętrzna stacja dyskow.

Grafika: rozdzielczość od 320x256 punktów (32 kolory) do 640x512 (16 kolorów). Dostępna paleta 4096 kolorów.

Klawiatura: 96 klawiszy wg normy DIN, 10 klawiszy funkcyjnych, oddzielne bloki klawiszy numerycznych i sterujących kursorem.

Amiga 2000 posiada wbudowany zegar i kalendarz z podtrzymaniem bateryjnym oraz możliwosc przyłączenia następujących kart:
- rozszerzająca RAM do 2 i 8MB z autokonfiguracją
- emulująca PC/XT z procesorem 8088 i stacją dysków 5 1/4" (360Kb)
- emulująca PC/AT z procesorem 80286 i stacją dyskow 5 1/4" (1.2Mb)
- karta kontrolera dysku twardego AMIGA dla dwóch dyskow ST506 i interfejs dla innych szybkich stacji dyskow i innych urządzeń peryferyjnych.



Amiga 500 i 100 to starsi bracia Amigi 2000. Z wyglądu przypominają C-128D. Amiga 500 posiada 512 Kb RAM, natomiast Amiga 1000 1Mb RAM (z możliwością zewnętrznego rozbudowania do 9Mb). Obydwa posiadają możliwość rozbudowy o karty emulujące pracę IBM PC/XT (cena takiej przyjemności przekracza jeszcze 1000 dolarów amerykańskich). Grafika i dźwięk zbliżone są do możliwości Amigi 2000.

Dominik Falkowski

PRZEDSTAWIAMY WARSAW BASIC (2)

PĘTLA INTERPRETERA

W poprzednim odcinku wyjaśniliśmy rolę pętli interpretera w wykonywaniu programu.

Pętla ta sprawdza, czy kolejny pobrany kod programu jest kodem instrukcji, separatora czy też końca linii. Jeżeli pętla pobrała kod inny od wyżej wymienionych, to sprawdza, czy jest to instrukcja podstawienia (LET), która powinna zaczynać się od nazwy zmiennej. Jak wiemy, nazwy zmiennych mogą zaczynać się tylko od litery (kody w systemie szesnastkowym od 41 do 5a). Jeżeli pobrany kod nie jest literą, to sygnalizowany jest błąd. I w tym miejscu można zmienić działalność pętli interpretera.

Jedną z możliwych dróg powiększenia listy instrukcji jest wstawienie do pętli, przed odesłaniem do wykonania domniemanej instrukcji podstawienia, sprawdzenia czy to nie jest jeden z wolnych kodów, reprezentujący jakiś znak specjalny.

Wybranymi znakami specjalnymi nie mogą być cyfry od 0 do 9 (taki znak nie mógłby być pierwszym znakiem po numerze linii, gdyż zostałby dopisany do tego numeru), litery od a do z (wtedy pętla interpretera odsyłałaby do domniemanej instrukcji podstawienia), ani żaden ze znaków Commodore, którego kod pokrywa się z kodem instrukcji (w systemie szesnastkowym od 80 do CA), ani też żaden ze znaków graficznych (poza π), znajdujących się na przedniej ściance klawiszy (te znaki dają się wpisywać do programu tylko w cudzysłowie lub komentarzu).

Kody funkcji standardowych i operatorów CBM V2.0 są na tym etapie interpretowania w zasadzie wolne, ale ich użycie jest niewygodne ze względu na niejednoznaczność tekstu programu.

Posługując się pozostałymi wolnymi kodami można zastosować włamanie „pierwszego stopnia" do pętli, polegające na sprawdzaniu kolejnych wolnych kodów, a po rozpoznaniu jednego z nich w programie, skierowaniu do działalności za nim ukrytej. W tym przypadku mamy do dyspozycji wszystkie wolne kody zarówno na etapie edycji jak i wyprowadzania programu. Mogą to być na przykład kody następujących znaków: ←, !, *, $, %, &, ', (,), +, −, £, @, •, ↑, [,] itd.

Włamanie „drugiego stopnia" polega na wybraniu jednego znaku specjalnego, sprawdzanego w pętli interpretera (pierwszy poziom interpretacji) i zagnieżdżeniu w pętli interpretera CBM V2.0 własnej pętli, interpretującej znaki bezpośrednio występujące po naszym wybranym znaku specjalnym (drugi poziom interpretacji). W ten sposób, zagnieżdżając coraz dalej można byłoby napisać interpreter który rozpoznawałby po znaku specjalnym dowolne nazwy instrukcji.

Takie włamanie pozwala w łatwy do zapamiętania sposób związać nazwę instrukcji z reprezentującym ją znakiem. W przypadku włamania „pierwszego stopnia" nie byłoby tej możliwości. Poza tym program pod kontrolą nowej pętli, który nie zawiera poza CBM V2.0 dodatkowych instrukcji, wykonuje się prawie tak samo szybko jak pod kontrolą pętli standardowej. Słowo „prawie" oznacza w tym przypadku, że każdy inny sposób ingerencji w pętlę daje w konsekwencji wydłużenie czasu interpretacji. Jeżeli przy włamaniu „drugiego stopnia" występują w programie nowe instrukcje, to program taki nie będzie wykonywany wolniej niż taki sam program z pętlą z włamaniem „pierwszego stopnia".

Jak już wspomniano, w przypadku pętli z włamaniem „drugiego stopnia" można byłoby używać więcej niż jednego znaku do rozpoznawania instrukcji na drugim poziomie interpretacji. Ale to spowodowałoby znaczne opóźnienie wykonania programu i zajmowałoby dużo miejsca w pamięci. Jeżeli chcemy posługiwać się dłuższymi nazwami instrukcji, to należałoby znacznie rozbudować procedury interpretujące program na etapie edycji i kodowania. Jest to zrobione w oryginalnych wersjach Warsaw BASIC-a. Tutaj tego nie będziemy rozbudowywać, ze względu na brak miejsca.

Włamania, o których tutaj mówimy, przewidziane przez projektantów interpretera CBM V2.0, nie będą miały nic wspólnego z jakże częstymi „włamaniami", polegającymi na usunięciu z programu nazwy firmy lub nazwisk autorów (w jednej z krążących, nielegalnie sprzedawanej na giełdach wersji Warsaw BASIC-a wstawiono zamiast nazwy programu i nazwisk autorów gwiazdki — to dopiero było włamanie!?).

W naszej Bajtkowej wersji w charakterze znaku specjalnego, rozpoznawanego na pierwszym poziomie interpretacji, wybraliśmy znak £ (funt). Znakami interpretowanymi na drugim (i wyższych) poziomie interpretacji, po znaku £, będą litery od a do z i znak @ (ucho słonia).

Włamania, o których wyżej była mowa, są możliwe dzięki „furtkom" pozostawionym w oryginalnym interpreterze CBM V2.0. Furtki te, nazywane wektorami, umożliwiają włączenie do procedur zapisanych w pamięci typu ROM własnych fragmentów kodu. Omawiana wyżej pętla interpretera zawiera następujący fragment:

$a7e1 jmp ($0308).

W normalnej konfiguracji pod adresem $0308/$0309, wpisany jest adres $a7e4 następnej instrukcji pętli interpretera, po instrukcji jmp($0308). Adres ten, wraz z kilkoma innymi kluczowymi dla pracy interpretera adresami, jest kopiowany z ROM do RAM przez procedurę inicjacyjną, podejmującą działanie bezpośrednio po włączeniu mikrokomputera do sieci. Dzięki temu, że adres ten znajduje się w RAM, możliwa jest nasza ingerencja, polegająca w pierwszej kolejności na zmianie tego adresu (porównaj linie 3-6 programu 1). Włączy to w pętlę interpretera procedurę rozpoczynającą się pod adresem $c40e (linie 11-33 programu 1). Jest to nasze włamanie „drugiego stopnia". Poziom pierwszy interpretacji realizują linie 11-17: pobranie kodu (11), rozpoznanie kodu £ (13), odesłanie na drugi poziom interpretacji (14) i powrót do pętli (16-17) w przypadku nie występowania znaku £.

Drugi poziom interpretacji realizują linie 18-33: pobranie kodu (19), rozpoznanie litery lub @ (20-21), odesłanie do błędu SYNTAX ERROR (23-24) i 25-33) pobranie adresu oraz odesłanie do wykonania instrukcji ukrytej pod tym adresem w przypadku wystąpienia litery. Opuszczone w wydruku programu 1 linie 34-59 zawierają adresy takich nowych procedur. W programie 2 wstawiono w tym miejscu $07 (bajt mniej znaczący — numer na stronie) i $af (bajt bardziej znaczący — numer strony). Pętla interpretera, w przypadku rozpoznania naszych kodów, odkłada ten adres na stos. Spowoduje to w dalszym działaniu odwołanie się do procedury umieszczonej w pamięci pod adresem $af08 (uwaga: $af08, a nie $af07!). W liniach 60-66 podano definicje etykiet występujących w tym programie.

Czytelnicy Bajtka, którzy pragną uczestniczyć wraz z nami w zabawie w rozbudowę interpretera i poznawać tajemnice Warsaw BASIC-a powinni teraz „wpałcować" program 2. Po bezbłędnym uruchomieniu tego programu można spróbować użycia n[illegible]ch instrukcji (od £@, £a do £z). Po wykonaniu jakiej[illegible] z tych instrukcji, nasz interpreter będzie sygnali[illegible] błąd SYNTAX ERROR. Co się stało? Czyżby ten rozbudowany interpreter nie rozumiał nowych słów? Interpreter nowe słowa rozumie, tylko każde z nich odsyła do tej samej procedury znajdującej się pod adresem $af08, pod którym interpreter ma zapisaną procedurę wyświetlającą napis SYNTAX ERROR i przerywającą wykonanie programu.

Dalsza rozbudowa naszego interpretera będzie zatem polegała na zmianie adresów w komórkach pamięci od $c431 do $c474 na takie, pod którymi będą się kryły procedury wykonujące zaplanowaną przez nas działalność i uzupełnianiu programu 2 o treść tych procedur.

Krzysztof Gajewski
Bogusław Radziszewski

PROGRAM 1

```
1              ORG    $C400
2  GETCHR      =      $73
3              LDA    #$0E
4              LDY    #$C4
5              STA    IGONE
6              STY    IGONE+1
7              NOP
8              NOP
9              NOP
10             RTS
11             JSR    GETCHR
12             PHP
13             CMP    #$5C
14             BEQ    HC01D
15             PLP
16             JSR    HA7ED
17             JMP    HA7AE
18 HC01D       PLP
19             JSR    GETCHR
20             AND    #$1F
21             CMP    #$1B
22             BCC    HC02C
23             LDX    #$0B
24             JMP    (IERROR)
25 HC02C       JSR    HC032
26             JMP    HA7AE
27 HC032       ASL
28             TAY
29             LDA    ADDRESS.Y
30             PHA
31             LDA    ADDRESS-1,Y
32             PHA
33             JMP    GETCHR

60 IERROR      =      $0300
61 IGONE       =      $0308
62 HA7AE       =      $A7AE
63 HA7ED       =      $A7ED
64 SYNTAX      =      $AF07
65 ADDRESS     =      $C440
66 CHROUT      =      $FFD2
```

PROGRAM 2

```
100 printchr$(147);" Interpreter 0.0 "
105 print:print" Czesc 1 "
110 x=50176:n=116:c=0
115 fori=0ton:reada
120 pokex+i,a:c=c+a:next
125 ifc=11160then135
130 print:print" Blad w czesci 1 ":end
135 sys12*4096+4*256
140 print:print" Czesc 1 ok "
145 data169,14,160,196,141,8,3,140,9,3,234,234
150 data234,96,32,115,0,8,201,92,240,7,40,32
155 data237,167,76,174,167,40,32,115,0,41,31,201
160 data27,144,5,162,11,108,0,3,32,50,196,76
165 data174,167,10,168,185,64,196,72,185,63,196,72
170 data76,115,0,7,175,7,175,7,175,7,175,7
175 data175,7,175,7,175,7,175,7,175,7,175,7
180 data175,7,175,7,175,7,175,7,175,7,175,7
185 data175,7,175,7,175,7,175,7,175,7,175,7
190 data175,7,175,7,175,7,175,7,175
```

NOWOŚCI

I NOWOSTKI

Dzięki panu Bolesławowi Chapińskiemu, który przesyła redakcji świeże cenniki i materiały mogę znów przedstawić Czytelnikom bieżące trendy w rozwoju oprogramowania i osprzętu do komputerów Commodore. Tym razem nowości jest rzeczywiście dużo...

Na pierwszy ogień ceny sprzętu w USA:

C-64C + GEOS	— 140 — 160 $
C-128	— 220 — 230 $
C-128 D	— 440 — 470 $
Stacja 1571	— 200 — 220 $
Stacja 1541C	— 140 — 160 $
Stacja 1581	— 180 — 220 $
Monitor 1902 (40/80 zn)	— 230 — 240 $
Monitor 1802 (40/80 zn)	— 150 — 180 $
Drukarka MPS 803C	— 110 — 140 $
Drukarka MPS 1200	— 200 — 230 $
Karta 1700 (128 KB)	— 100 — 120 $
Karta 1750 (512 KB)	— 150 — 180 $
AMIGA 1000	— 650 — 800 $
AMIGA 1000 R	— 450 — 550 $
Mysz	— 30 — 45 $
PC 10 (odpowiednik IBM XT)	— 450 — 550 $
PC 10 + 20 MB hard disk	— 650 — 750 $
Dysk twardy do C-64	— 900 $
Dysk twardy do C-64 i 128	— — 950 $

Na największą uwagę zasługuje tu moim zdaniem cena AMIGI 1000 R oraz komputerów w pełni profesjonalnych PC-10 (i pokrewnych do PC-40 włącznie). Biorąc pod uwagę cenę dysku twardego do C64 i 128 okazuje się, ze za te same pieniądze można kupić dwa komputery typu XT lub dwie Amigi 1000 R. Te same ceny w zestawach wyglądają znacznie korzystniej. Przykładowo C64C + 1541C + monitor mono + kolorowa drukarka są dostępne już za 348 $, sam C-64 kupowany z określoną ilością oprogramowania można dostać za 90 $, drukarkę MPS 803 za 110 $.

*

Firma Xetec opracowała nareszcie wersję znakomitego programu FONTMASTER dla C-128 (sam program zostanie przedstawiony w dodatkowym numerze BAJTKA poświęconym Commodore). Na dyskietce zapisano 56 gotowych do użytku czcionek oraz program do projektowania własnych zestawów znaków. Program może współpracować z ponad 110 drukarkami do których programy sterujące są także zapisane na dyskietce. Oprócz tego na drugiej stronie dyskietki znajduje się program SPELL CHECKER (kontrolujący ortografię i podający synonimy) zawierający ponad 102 tysiące słów. Dyskietka nie jest zabezpieczona przed kopiowaniem, jednakże program działa tylko wraz z odpowiednim urządzeniem zabezpieczającym („dongle"). W sprzedaży znajdzie się też niebawem dyskietka z dodatkowymi czcionkami. Cena programu — 70 $.

Ta sama firma opracowała nowy model twardego dysku o pojemności 20 MB dla C-64 i C-128. System o nazwie Lt. KERNAL składa się z kontrolera, dysku oraz specjalnej karty przyłączanej do portu rozszerzania (expansion port) w komputerze. DOS zawiera ponad 45 rozkazów wśród których istnieje również opcja FREEZE. Opcja ta umożliwia użytkownikowi przenoszenie zabezpieczonych programów na dysk twardy, jednakże bez możliwości późniejszego ich przenoszenia na dyskietki (program nie będzie działał). Główną wadą jest tu niestety wysoka cena urządzenia.

*

Na rynku ukazała się trzecia z kolei wersja karty znanej jako FINAL CARTRIDGE — FINAL CARTRIDGE III. Z ciekawszych zastosowań warto wspomnieć o poprawieniu szybkości wczytywania programów z dyskietki (25 razy szybciej niż 1541), wprowadzeniu procesora tekstu (!), zegara, kalendarza i kalkulatora. Ponadto FINAL CARTRIDGE III oferuje użytkownikowi ponad 24 KB RAM więcej dla użytkownika, osławiony „freezer" umożliwiający zatrzymanie dowolnego programu czy zapis całej pamięci na taśmie/dysku, opcję DISK APPEND umożliwiającą dopisywanie danych do programów już na dysku istniejących i wiele innych. Cena modułu — 70 dolarów.

*

Amatorzy gier dostaną niebawem do rąk MASTER NINJA (dla C-64) — program sprzedawany przez firmę Paragon Software. Tematem jest magiczny miecz skradziony przez złego demona, który należy odzyskać przeszukując zamek wspomnianego demona. Z kolei Broderbund Software zapowiada dwa tytuły — SUPERBIKE CHALLENGE (wyścigi motocyklowe — do wyboru 12 znanych na świecie torów), oraz MAGNETRON (gra kosmiczna — należy zniszczyć 50 baz wojennych kosmitów najeźdźców). Ciekawostką jest tu fakt, że promocja SUPERBIKE CHALLENGE będzie powiązana z konkursem, w którym główną nagrodą jest motocykl SUZUKI o pojemności 650 cm^3...

*

Firma Scinort Micro opracowała nową kartę rozszerzającą pamięć dla C-64. Karta ta o pojemności 32 KB zawiera podtrzymywanie bateryjne (program zapisany może być w niej przechowywany przez ponad rok) oraz opcje do automatycznego wczytywania i uruchamiania programów w BASIC lub w języku maszynowym, łączenia programów ze sobą lub ich kasowania. Cena ok. 60 $.

*

INSTANT MUSIC to nowy program muzyczny firmy Electronic Arts. Jedną z jego największych zalet jest możliwość współpracy poprzez interface MIDI z profesjonalnymi elektronicznymi instrumentami muzycznymi. Program jest łatwy w obsłudze zarówno dla muzyków jak też i początkujących i zawiera na dyskietce 40 zapisanych utworów.

*

Choć telewizja satelitarna dopiero w kraju raczkuje, to dostępny jest już program SATTCOM-64 opracowany przez Strategic Marketing Resources. Jest to uzupełnienie innych programów radioamatorskich pozwalające na obliczenie pozycji satelity telekomunikacyjnego. Informacja podawana przez program pozwala na odpowiednie ustawienie anteny satelitarnej na odbiór jednego z 15 satelitów.

*

CHUCK YEAGER'S ADVANCED FLIGHT TRAINER będzie niewątpliwie łakomym kąskiem dla miłośników lotnictwa. Program ten produkcji Electronic Arts jest prawdziwym trenazerem lotu i umożliwia naukę latania na 14 różnych typach samolotów, m.in. P-51 MUSTANG, SOPWITH CAMEL czy BELL-X1. Z kolei trójwymiarowy symulator lotu ECHELON oferowany przez firmę Access umożliwia latanie samolotem Lockheed C-104 Tomahawk.

*

Na półkach księgarskich ukazała się już książka COMPUTE!'s COMMODORE BUYER'S GUIDE będąca encyklopedią i przewodnikiem dla posiadaczy Commodore. Omówiono tam i przedstawiono olbrzymią liczbę programów wszelakiego rodzaju (wraz z adresem producenta i ceną) oraz podstawowe urządzenia peryferyjne takie jak drukarki, drążki sterowe, stacje dysków itp.

*

Firmy sprzedające komputery używane przechodzą już same siebie. Jeżeli zdecydujesz się przykładowo na zakup C-128, stacji 1571 i monitora 1802 to możesz także kupić AMIGĘ 2000 jeśli dopłacisz 970 $, AMIGĘ 500 jeśli dopłacisz 159 $, C-128 D jeśli dopłacisz 95 $. Pozostałe ceny są także bardzo atrakcyjne (np. za 79 $ stacja 1581, gdy kupujesz ją wraz ze stacją 1571). Brać i wybierać...

*

BerkeleySoft, firma która opracowała znakomity program GEOS, znany szeroko na całym świecie, nadal uzupełnia bibliotekę oprogramowania tego systemu. Oprócz GEOFILE, DESKPACK, GEODEX, GEOPUBLISH, GEOCALC i kilku innych wprowadziła na rynek nowy zestaw przeznaczony dla C-128. W zestawie tym znajdują się programy GEOS 128, DESKPACK 128, GEOFILE 128 oraz GEOCALC 128. Oprócz tego na rynku ukazał się także nowy produkt — GEOPROGRAMMER będący w rzeczywistości pakietem do programowania w języku maszynowym. Program składa się z trzech głównych podprogramów — geoAssembler, geoLink oraz geoDebugger. Cena programu 70 dolarów. Inny producent oprogramowania, firma Timeworks, Inc. twórca znakomitych programów takich jak DATA MANAGER, SWIFTCALC, WORD WRITER, FINANCIAL PLANNER czy PARTNER 128 wprowadziła na rynek nowy program DESKTOP PUBLISHER. Jest to program umożliwiający projektowanie własnej gazetki, papeterii, druków firmowych itp. DESKTOP PUBLISHER ma możliwość odczytywania rysunków wykonanych pod PRINT MASTER, PRINT SHOP i GEOWRITE, współpracuje ponadto ze zbiorami tworzonymi za pomocą edytorów tekstu WORD WRITER, PAPERCLIP oraz WORD PRO. W sprzedaży znajduje się też ulepszona wersja programu PRINT MASTER — PRINT MASTER PLUS.

*

Dwa nowe zastosowania profesjonalne pochodzą z kolei z firmy SPINNAKER. Pierwszym z nich jest znany edytor tekstu WORD-PRO, w którym dodano procedury szybkiego wczytywania/zapisu danych, drugim zaś POWER C — wersja języka C dla C-64.

Opracował: Klaudiusz Dybowski

OD ŚRODKA cz. III

Po przeczytaniu ostatniego odcinka, wczytanie do pamięci programu tak, by się nie zdążył uruchomić, nie powinno być dla Ciebie żadnym problemem, choć wczytany program wcale nie musi wyglądać „normalnie".

Na przykład w programie jest linia o numerze zero albo linie są uporządkowane z malejącymi numerami, nie można wykonać **EDIT** dla żadnej linii, widać podejrzaną instrukcję **RANDOMIZE USR 0** lub po prostu nic nie widać, bo program nie daje się wylistować. Jeśli w programie, do którego się włamujesz, zauważyłeś coś dziwnego, to najlepiej obejrzyj go w trochę inny, niż normalny, sposób — nie za pomocą **LIST,** lecz bezpośrednio — używając funkcji **PEEK**

Najpierw jednak musimy się dowiedzieć, w jaki sposób jest umieszczany w pamięci tekst programu w BASIC-u. Program składa się z kolejnych linii i tak też jest przechowywany w pamięci.

A oto, jak wygląda pojedyncza linia programu (rys. 1)

Zajmuje ona co najmniej 5 (a właściwie 6, bo tekst nie może być pusty) bajtów. Pierwsze dwa oznaczają jej numer, lecz uwaga! — odwrotnie niż wszystkie dwubajtowe liczby zapisane w pamięci — tutaj pierwszy bajt jest bardziej znaczący (MSB — Most Signifficant Byte), a drugi mniej znaczący (LSB — Less Signifficant Byte). Jeśli więc będą one przykładowo równe, pierwszy — 0, drugi — 10, to nie będzie to oznaczało 2560 (0 + 10 ✱ 256), lecz 10 (256 ✱ 0 + 10).

Następne dwa bajty to długość linii, tzn. ile znaków zawiera tekst linii, wraz z kończącym go znakiem **ENTER** (heksadecymalnie OD). Za tymi bajtami znajduje się już właściwy tekst linii, zakończony przez **ENTER.** Jeśli wpiszemy np. taką linię:

10 REM BASIC

i wyślemy ją wciskając klawisz **ENTER**, to w pamięci zostanie ona zapisana jako ciąg bajtów (rys. 2)

Parametr „długość linii" dotyczy jedynie jej tekstu, więc chociaż cała linia zajmuje w pamięci 11 bajtów, to parametr ten wskazuje tylko na 7 bajtów: 6 bajtów tekstu i siódmy — znak **ENTER** kończący linię.

Rozumiesz już chyba, na czym polega stosowany często trick z linią o numerze zero. Wystarczy w dwa pierwsze bajty linii wpisać liczby 0 (za pomocą **POKE**-ów), by linia ta stała się linią zerową. Jeśli chcemy zmienić numer pierwszej linii w programie (a nie są podłączone interfejsy żadnej szybkiej pamięci masowej, bo wówczas zmienia się adres początku BASIC-a), to wystarczy wpisać

POKE 23755,x: POKE 23756,y

a linia otrzyma numer 256 ✱ x + y. Niezależnie jednak od tego, jaki on jest — pozostanie ona w pamięci tam, gdzie była. Jeśli więc wpiszemy np.

```
10  REM linia nr 10
20  REM linia nr 20
```

```
POKE 23755,0: POKE 23756,30
```

to pierwsza linia w programie otrzyma numer 30, pozostanie jednak w pamięci jako pierwsza, a na ekranie uzyskamy wydruk:

```
30  REM linia nr 10
20  REM linia nr 20
```

Aby więc zacząć odbezpieczać program, w którym występują linie zerowe lub odwrotnie uporządkowane, należy poszukać adresów początków poszczególnych linii i w ich polu „nr linii" umieszczać kolejno np. 10, 20, 30... W pamięci linie znajdują się jedna za drugą, więc z odnalezieniem ich początków nie powinieneś mieć kłopotu. Jeżeli x wskazuje adres jakiejś linii programu, to adres następnej jest równy:

x + PEEK (x + 2) + 256 ✱PEEK (x +3) +4

— do adresu linii dodajemy długość jej tekstu zwiększoną o 4 bajty, bo tyle zajmują parametry „nr linii" i „długość linii".

Taka metoda znajdowania początków linii nie skutkuje niestety, gdy zastosowane jest drugie zabezpieczenie — fałszywa długość linii. Polega ono na tym, że w polu „długość linii", zamiast prawdziwej wartości podana jest bardzo duża liczba — rzędu 43 — 65 tysięcy. Zabezpieczenie to jest bardzo często stosowane, gdyż zazwyczaj uniemożliwia wczytanie programu przez **MERGE** (czyli tak, by się nie uruchomił). Dzieje się tak dlatego, że **MERGE** ładuje program z taśmy w obszar **WORKSPACE,** da następnie interpreter analizuje cały wczytany program linia po linii: sprawdza kolejno numer każdej z nich, a następnie umieszcza ją w odpowiednim miejcu obszaru przeznaczonego na tekst programu w BASIC-u. Na linię tę musi przygotować tam odpowiednią ilość wolnych bajtów, „rozsuwając" już istniejący tekst programu. Jeśli w polu „długość linii" podana będzie bardzo duża wartosc, to interpreter będzie usiłował zrobic właśnie tyle bajtów miejsca w obszarze tekstu programu w BASIC-u co skończy się komunikatem „Out of memory" lub po prostu zawieszeniem się systemu. Aby wczytać taki program nie powodując jego uruchomienia, należy użyć odpowiedniego włamywacza, np takiego, jak przedstawiony miesiąc temu program „load/merge".

Dodatkowym skutkiem podania fałszywej długości, jest niemożność poprawiania takiej linii przez ściągnięcie jej do pola edytora klawiszem EDIT. Sytuacja przedstawia się podobnie: system operacyjny usiłuje zrobić miejsce na tę linię w obszarze edycji linii BASIC-a (od zmiennej **E LINE** do **WORKSP** patrz rys 2 w części I). Wymaga to jednak zbyt dużej ilości wolnej pamięci, więc kończy się to tylko ostrzegawczym dźwiękiem

Jeżeli program zabezpieczony jest w ten sposób, trzeba adresów kolejnych linii szukać „ręcznie" lub domyślać się, gdzie one są, pamiętając o tym, że każda linia kończy się znakiem ENTER (ale nie każda liczba 13 oznacza **ENTER**.

Aby przeglądać program w BASIC-u, wpisz taką linię (listing C)

```
FOR n=23755 TO PEEK 23627+256*PEEK 23628
: PRINT n;"    ";PEEK n,CHR$ PEEK n AND P
EEK n>31: NEXT n
```

Wydrukuje ona kolejno: adres, zawartość bajtu o tym adresie oraz znak o tym kodzie, jeśli tylko nie jest to znak kontrolny (tzn. o kodzie 0–31)

Po zmianie numeracji linii i oszukiwaniu długości linii, następnym sposobem zabezpieczania programów są znaki kontrolne, uniemożliwiające najczęściej prawidłowe wylistowanie programu, choć nie tylko.

Wróćmy do pierwszego przykładu (linia **„10 REM BASIC").**

Tekst linii składał się z siedmiu znaków — słowa kluczowego REM" oraz pięciu liter i znaku **ENTER** Tak dzieje się zawsze, jeśli w linii znajduje się in-

MSB LSB	LSB MSB		
2 bajty	2 bajty	...	ODh
numer linii	długość tekstu + ENTER	tekst	znacznik końca – ENTER

Rys. 1

0	10	7	0	234	66	65	83	73	67	13
10		7		REM	B	A	S	I	C	ENTER

linia numer 10 — długość linii 7 bajtów — t e k s t l i n i i

Rys. 2

0	10	18	0	246	49	48	14	0	0	10	0	0	44	57	14	0	0	9	0	0	13
nr 10		dług 18		PLOT	1	0	numer: 10						,	9	numer: 9						ENTER

Rys. 3

strukcja REM — wszystkie znaki wpisane z klawiatury, znajdujące się za tą instrukcją zostaną umieszczone w tekście linii bez najmniejszych zmian. Inaczej jednak przedstawia się sytuacja, gdy w linii znajdują się inne instrukcje, wymagające parametrów liczbowych (a tak jest zazwyczaj). Wpiszmy np. linię:

10 PLOT 10,9

i zobaczmy w jaki sposób została zapisana w pamięci (najlepiej — wpisując podaną wyżej linię **FOR n=23755 TO...**). Wygląda ona w taki sposób, jak na rys. 3.

Jak widać tekst zotał zmodyfikowany — po ostatniej cyfrze każdej liczby występującej w tekście linii jako parametr, interpreter zrobił 6 bajtów miejsca i umieścił tam znak o kodzie 14 oraz pięć bajtów, w których zapisana jest wartość tej liczby, ale w sposób zrozumiały dla interpretera. Przyspiesza to w pewnym stopniu działanie programów w BASIC-u, ponieważ podczas działania programu interpreter nie musi za każdym razem przeliczać liczby z postaci alfanumerycznej (tzn. ciągu cyfr) na pięciobajtową postać umożliwiającą wykorzystanie jej do obliczeń, lecz gotową wartość pobiera z pamięci, zza znaku kontrolnego CHR$ 14. Ten podwójny zapis daje także duże możliwości utrudniania dostępów do programów. W wielu programach ładujących (tzw. ładowaczach lub loaderach) występuje taka linia:

O RANDOMIZE USR 0: REM ...

Na pierwszy rzut oka — po uruchomieniu się, program ten powinien wykasować całą pamięć, tak się jednak nie dzieje. Po dokładniejszym obejrzeniu (przez PEEK — linią FOR n= 23755 ...) okazuje się, że po USR 0 i znaku CHR$ 14 wcale nie ma pięciu zer (bo tak w pięciobajtowym zapisie wygląda liczba zero)**, lecz np. 0, 0, 218, 92, 0, co jest równoznaczne liczbie 23770. Funkcja USR nie skacze więc pod adres 0, lecz właśnie 23770, a jest to adres bajtu znajdującego się zaraz za instrukcją REM w naszym przykładzie. Tam zwykle znajduje się program ładujący napisany w języku maszynowym.

Następnym znakiem kontrolnym, często stosowanym w różnych zabezpieczeniach, jest **CHR$ 8** — „backspace", czyli spacja do tyłu. Wydrukowanie tego znaku powoduje cofnięcie pozycji wydruku o jeden znak w lewo. Można więc za jego pomocą zakrywać niektóre instrukcje na listingu, drukując w ich miejscu inny tekst. Jeśli np. w pamięci znajdują się kolejno znaki:

LET a=USR 0: ←←←←←←←←←←←←←
LOAD "": ...

(← oznacza CHR$ 8), to instrukcja LOAD "" i dalszy tekst zakryją wcześniejszą instrukcję LET a= USR 0. Chociaż na listingu widoczna jest tylko instrukcja LOAD "", to dalsza część programu nie jest ładowana przez nią, lecz przez program maszynowy uruchamiany funkcją URS 0 (co nie musi oczywiście oznaczać skoku pod adres 0). Takie zabezpieczenie jest np. stosowane w loaderze programu Beta Basic 1.0.

To by było wszystko na dzisiaj, choć znaków kontrolnych jest oczywiście więcej. Ich opis dokończymy w następnym odcinku.

Tomasz Surmacz
Robert Dudzik

*) Wszystkie słowa kluczowe — instrukcje i funkcje, a także znaki <=, >= i <> mają jednobajtowe kody z zakresu 165 — 255. Jeśli nie wiesz jakie — wpisz:
FOR n=165 TO 255: PRINT n,CHR$ n: NEXT n

**) Jeśli jest to liczba całkowita z zakresu -65536 do 65535, to w zapisie pięciobajtowym wygląda ona następująco:
pierwszy bajt — 0
drugi bajt — 0 dla liczby dodatniej, 255 dla ujemnej
trzeci i czwarty — kolejno: młodszy i starszy bajt liczby (lub liczby + 65536, jeśli była ujemna)
piąty bajt — 0

SZUKAMY LICZBY
CZYLI SUKCESYWNA APROKSYMACJA

W instrukcji do Spectrum znalazłem następujący program:

```
10 LET a=INT (RND*100)
20 LET a$=""
30 INPUT (a$) "Zgadnij liczbe ",b
40 IF a=b THEN PRINT "Gratuluj[illegible]": STOP
50 [illegible] THEN LET a$="[illegible]"
60 IF [illegible] THEN LET a$="Za duz[illegible]"
[illegible]
```

Jest to doskonała zabawa na prawie cały dzień. Zgadujemy tu liczbę pomyślaną przez komputer. Najefektywniejszą taktyką odgadywania jest dzielenie rozpatrywanego przedziału na pół. Na przykład jeżeli zgadujemy liczbę z przedziału (0,100), podajemy 50. Jeśli jest ona za mała, bierzemy przedział (50,100), podajemy 75 i tak dalej. W zasadzie ósmy strzał powinien być trafiony.

Metoda kolejnych przybliżeń to właśnie aproksymacja. Powinna on być sukcesywna — dawać rezultat jak najszybciej.

Spróbujmy odwrócić sytuację — my pomyślimy liczbę, a komputer będzie ją odgadywał

```
[illegible] LET y=100
20 LET c=INT [illegible]
30 INPU[illegible] a$
40 IF a$=[illegible] THEN LET [illegible] GO [illegible]
50 IF a$=[illegible] THEN LET y=c [illegible]
60 IF a$="=" THEN STOP
70 GO TO 20
```

W odpowiedzi na pytanie „Czy xx" należy podać jeden ze znaków — ">", "<" lub "=". Rozpatrujemy przedział (0,100).

Istnieje niejedna metoda aproksymacji. Rozważmy program:

```
10 DIM a(6)
15 LET a(1)=1
20 FOR i=2 TO 6: LET a(i)=2[illegible]
[illegible] NEXT i
[illegible] LET [illegible]
40 FOR [illegible] TO 1 STEP -1
50 [illegible] "Czy [illegible]",a$
60 [illegible] a$[illegible] AND a$[illegible] AND
a$[illegible] THEN GO TO 50
70 IF a$="[illegible]" THEN LET c=c+a(i)
80 IF a$="[illegible]" THEN LET c=c-a(i)
90 IF a$="=" THEN GO TO 110
100 NEXT
110 [illegible] "Liczba=";c
```

W tablicy a zapisaliśmy sześć kolejnych potęg dwójki. Najwyższa liczba to 64, więc kompuer odgadnie tu liczbę z przedziału (0 127). Jako pierwsza podawana jest liczba 64. Jeżeli pomyślana jest większa, do 64 dodawana jest 32, jeśli mniejsza, to 32 jest odejmowane i tak dalej. Tu maksymalna ilość pytań jest także 8 ze względu na włączenie zera do rozpatrywanego przedziału.

Możemy rozszerzyć tablicę na przykład do 16 pozycji; trzeba wtedy odpowiednio pozmieniać początki i końce pętli, lecz uzyskamy zakres do 131069, a maksymalną ilością pytań będzie najwyższa potęga dwójki, czyli 16.

W zasadzie nie można powiedzieć, która z metod jest najlepsza. W pamięci lepiej używać metody dzielenia przedziałów na pół, maszyna szybciej aproksymuje dwójkowo.

Marcin Przasnyski

Bzzz INACZEJ

Po przeczytaniu opisu gry „Bzzz" w numerze 3/87 „Bajtka" postanowiłem napisać ją w BASIC-u. Program okazał się prosty i o wiele krótszy od wersji w kodzie maszynowym. Mimo to jest on dosyć szybki.

W linii 2 po pierwszym słowie „GRACZ" należy wstawić zieloną, a po drugim niebieską kratkę. Zapalenie się czarnego kwadracika informuje o przegranej któregoś z graczy.

W liniach 9500 i 9600 musi być czarna spacja graficzna.

Hubert Sobczak

```
   1 CLS : LET g=1: LET h=1: LET
q=100: LET w=50: LET e=170: LET
r=50
   2 PRINT AT 21,0;"GRACZ
        GRACZ    "
   5 FOR i=0 TO 255: PLOT i,9: P
LOT i,175: NEXT i
   6 FOR i=9 TO 175: PLOT 0,i: P
LOT 255,i: NEXT i
   7 PRINT PAPER 4;AT 15,12;" ":
 PRINT PAPER 1;AT 15,21;" "
  10 LET a=IN 63486: LET b=IN 61
438
  20 IF a=190 THEN LET g=4
  30 IF a=183 THEN LET g=2
  40 IF a=187 THEN LET g=1
  50 IF a=189 THEN LET g=3
  60 IF b=183 THEN LET h=4
  70 IF b=190 THEN LET h=2
  80 IF b=187 THEN LET h=1
  90 IF b=189 THEN LET h=3
 100 IF g=4 THEN LET q=q-1
 110 IF g=2 THEN LET q=q+1
 120 IF g=1 THEN LET w=w+1
 130 IF g=3 THEN LET w=w-1
 140 IF h=4 THEN LET e=e-1
 150 IF h=2 THEN LET e=e+1
 160 IF h=1 THEN LET r=r+1
 170 IF h=3 THEN LET r=r-1
 180 IF POINT (q,w)=1 THEN PLOT
FLASH 1;q,w: GO TO 9500
 190 IF POINT (e,r)=1 THEN PLOT
FLASH 1;e,r: GO TO 9600
 200 PLOT q,w: PLOT e,r: GO TO 1
0
9000 FOR i=0 TO 30: BEEP .008,i:
 NEXT i: FOR i=0 TO 500: NEXT i:
 GO SUB 9998: RUN
9500 PRINT AT 21,9;"█": GO TO 90
00
9600 PRINT AT 21,26;"█": GO TO 9
000
9998 LET a$=INKEY$: IF a$<>"" TH
EN RETURN
9999 GO TO 9998
```



PLOTER DXY-980A

Testowany ploter DXY-980A firmy ROLAND DG został dostarczony do redakcji przez zachodnioniemiecką firmę ABC DATA. Jest on seryjnie produkowanym ploterem i stanowi reprezentatywną próbkę wyrobów firmy ROLAND. Niniejszy test jest wyłącznie „testem użytkownika” tzn. badaniu podlegała tylko strona użytkowo-funkcjonalna, nie zaś rozwiązanie układowe, wytrzymałość mechaniczna (np. badania na wstrząsarce) itp.

OBUDOWA

Ploter ma blaszaną obudowę W prawym górnym rogu umieszczony jest panel sterujący wraz z wyświetlaczami pokazującymi aktualną pozycję głowicy we współrzędnych X-Y. W obudowie zamontowany jest zespół 8 uchwytów dla pisaków, z których pi[illegible]ki pobierane są automatyc[illegible]. Dodatkowo obok uchwytów [illegible] otworów służących do przechowyw[illegible] [illegible] pisaków w cz[illegible] [illegible] są uż[illegible]wane. Otwory te [illegible] aktowane aby stanowiły szczeln[illegible] zamknięcie dla pisaków, zabezpieczając je przed wysychaniem. Obudowa zost[illegible] tak skonstruowana aby ploter mógł pracować w pozycji poziomej (położony na stole) lub w pozycji ukośnej (specjalna podpórka ruchoma). Jest to szczególnie istotne ze względu na ilość miejsca jaką mozemy przeznaczyc na ustawienie plotera. W pozycji ukosnej zajmuje on ok 40 proc miejsca potrzebnego do ustawienia go w pozycji poziomej.

ZASILANIE

[illegible] wyłącz[illegible] [illegible]nia umie[illegible] [illegible] z tyłu obudow[illegible] [illegible] rozwiązanie, [illegible] przy [illegible]nym [illegible] [illegible] klawisza. Ploter zasilany jest z oddzielnego wolno stojącego zasilacza sieciowego Rozwiązanie takie, acz korzystne z technicznego punktu widzenia (wydzielenie źródła ciepła i pola magnetycznego z obudowy plotera), jest bardzo niewygodne dla użytkownika, gdyż wyłącznik zasilania odłącza ploter a zasilacz pozostaje nadal pod napięciem. Zmusza to użytkownika do wyjmowania wtyczki z gniazdka sieciowego lub do robienia dodatkowego wyłącznika na przewodzie zasilacza.

PULPIT STEROWNICZY

Pulpit sterowniczy składa się z 10 klawiszy pozwalających na szybkie i wolne przesuwanie głowicy, ustawienie punktów bazowych wyznaczających położenie i rozmiar papieru, podnoszenie pisaka i wstrzymanie transmisji podczas pracy. Dodatkowo przy wyswietlaczach znajdują się dwa przyciski zerowanie wyświetlaczy i włączanie elektrostatycznego trzymania papieru. Sposób rozwiązania pulpitu jest bardzo ergonomiczny a co za tym idzie wygodny dla użytkownik[illegible]

PAPIER

Jedną z najbardziej istotnych zalet tego plotera jest sposób trzymania papieru. W modelu tym zastosowano metodę elekrostatycznego trzymania papieru. Wystarczy ułożyć arkusz na blacie plotera i przycisnąć klawisz „paper hold”. Elektrostatyczne trzymanie papieru jest nieporównywalnie wygodniejsze od metody pasków magnetycznych. Efektywna siła przylegania papieru jest bardzo duża. Trzeba również dodać, że mimo konieczności wytworzenia dużego ładunku elektrostatycznego jest to metoda absolutnie bezpieczna. Dotykanie blatu plotera nie pociąga za sobą żadnych skutków (wstrząs elektryczny) dla człowieka. Maksymalny rozmiar papieru a co za tym idzie maksymalny rozmiar rysunku jaki można wykonać odpowiada formatowi A-3

PODŁĄCZENIE DO KOMPUTERA

Ploter posiada wbudowane dwa łącza: równoległe typu Centronics oraz szeregowe typu RS-232. Są to najbardziej popularne łącza w klasie komputerów osobistych. Umożliwiają one podłączenie plotera do dowolnego komputera klasy „Personal”. Łącze RS-232 zapewnia pełną gamę szybkości (od 50 do 9000 bodów), możliwe jest ustawienie długości słowa 8 bitów /7 bitów oraz bitów parzystości (EVEN, ODD, NONE). Możliwe są cztery protokóły współpracy: sprzętowy, programowy, X on/X off oaz ENQ/ACK. Wszystkie parametry pracy oraz jego wybór ustawione są przy pomocy mikroprzełączników umieszczonych z tyłu obudowy plotera. Ploter był podłączany do komputera IBM PC XT/AT przez oba typy łączy. W przypadku łącza RS-232 ploter pracował ze wszystkimi szybkościami transmisji i nie stwierdzono żadnych problemów współpracy.

PODSTAWOWE DANE FUNKCJONALNE

Jak na sprzęt popularny (niska cena) ploter zapewnia bardzo dobre parametry rysunku. Przy dużej szybkości rysowania (230 mm/sek.)zapewniona jest bardzo wysoka rozdzielczość (dokładność pozycjonowania głowicy 0.05 mm/krok). Ploter zapewnia automatyczne skalowanie prowadzone zgodnie z aktualnie ustawionym formatem papieru (ustawianym przy pomocy mikroprzełączników). Oczywiście aktualny rozmiar rysunku można ustawić definiując z pulpitu sterowniczego punkty bazowe P1 i P2, odpowiadające dolnemu lewemu rogowi rysunku i górnemu prawemu rogowi rysunku. O jakości skalowania może świadczyć fakt wykonania rysunku formatu A-2 na papierze formatu A-4 Wszystkie proporcje zostały idealnie zachowane.

Ploter pozwala na wybranie jednego z osmiu zestawów znaków. Użytkownik ma do dyspozycji następujące zestawy: ANSI ASCI ANSI ASC II, francusk[illegible] niemiecki, hiszpańsk[illegible] skandynaw[illegible] szwedzk[illegible] duński i norweski

TRYBY PRACY

Self test — pozwala na sprawdzenie poprawności działania plotera bez podłączania do komputera.

Printer — ploter pracuje jak drukarka i może być używany np. do listowania programów. W modzie tym jest jednak dużo wolniejszy od drukarki.

DXY — ploter przyjmuje standardowe komendy języka graficznego DXY-GL. Jest to jeden z najbardziej popularnych języków tego typu pozwalający na wykorzystanie plotera z większością istniejących programów graficznych.

RD-GL — w trybie tym ploter akceptuje komendy rozszerzonego języka graficznego RD-GL firmy ROLAND. Język ten zawiera wiele dodatkowych funcji graficznych np. komendy dotyczące okien i skalowania.

Dodatkowo ploter pozwala na włączenie lub wyłączenie automatycznego skalowania oraz pracy z buforem wewnętrznym plotera. Wyprowadzono również specjalny „monitor mode" pozwalający na kontrole pracy plotera. Można go jednak tylko wykorzystać przy pracy plotera z łączem RS-232. Bardzo istotną cechą jest możliwość wykorzystania plotera jako digitizera. Jednak aby to było możliwe konieczne jest podłączenie plotera tak jak z „monitor mode" przez RS-232.

PISAKI

Jedną z istotnych wad plotera są pisaki. Standardowo dostarczone pisaki nie zapewniają jednostajnej grubości linii przy różnych szybkościach oraz ulegają dość szybkiemu wyczerpaniu. Rozwiązaniem jest kupienie specjalnego zestawu pisaków firmy STETHLER, które zapewniają bardzo wysoką jakość rysunku. (niestety za dużo wyższą cenę).

WSPÓŁPRACA Z OPROGRAMOWANIEM

Omawiany ploter był testowany w podłączeniu do komputera IBM PC i współpracował bez jakichkolwiek problemów z programami ORCAD, SMARTWORK i REDAC

PODSUMOWANIE

Wydaje się, że w swojej klasie ploterów popularnych DXY-980A jest egzemplarzem bardzo udanym. Niska cena i solidność wykonania to podstawowe zalety tego modelu.

Zalety:
— elektrostatyczne trzymanie papieru
— duża szybkość pracy
— duża dokładność pracy
— solidna i funkcjonalna obudowa
— możliwość wykorzystania plotera do digitalizacji
— łącze RS-232 i Centronics

Wady:
— pisaki
— wydzielony zasilacz sieciowy
— dość długi czas utrzymywania ładunku trzymającego papier po wyłączeniu klawisza „paper hold".

Sławomir Polak

KLAN AMSTRAD-SCHNEIDER

PCW 9512



Komputery firmy AMSTRAD zyskały sobie bardzo dobrą opinię dzięki niskiej cenie za sprzęt dobrej jakości. Oprócz modeli 16-bitowych, zgodnych z IBM PC, firma kontynuuje także produkcję maszyn 8-bitowych, opartych o wysłużony, ale ciągle jeszcze popularny mikroprocesor Zilog Z80.

Najbardziej znanym przedstawicielem tej rodziny jest CPC 6128, cieszący się u nas wielkim powodzeniem. Znacznie mniej docenione, mimo swoich niewątpliwych zalet, są komputery serii PCW: 8256 i 8512. Maszyny te przeznaczone do prac biurowych, dostarczane są razem z edytorem tekstu o nazwie Locoscript. W naszych warunkach najbardziej istotna jest możliwość wykorzystania ich do pracy w systemie operacyjnym CP/M Plus, który otwiera dostęp do dużej ilości oprogramowania narzędziowego i aplikacyjnego: języki programowania, bazy danych, programy graficzne itp.

Najnowszym produktem AMSTRADA w tej serii jest PCW 9512. Najważniejszą zmianą w stosunku do poprzedników jest zastosowanie drukarki rozetkowej (daisywheel), dającej znacznie lepszą jakość druku niż drukarki mozaikowe (dot matrix printer). Wadą tego rozwiązania jest całkowity brak możliwości graficznych, mała szybkość druku i kłopoty z implementacją alfabetów narodowych. O ile dotychczas większość zmian tego typu ograniczała się do programowej lub sprzętowej wymiany generatora znaków, to tutaj potrzebna jest odpowiednia rozetka do drukarki. Kolejną różnicę stanowi zastosowanie dwustronnej stacji dysków o dużej pojemności 720 KB. Jest to krok naprzód, ponieważ poprzednio używany napęd o pojemności 2x180 KB utrudniał pracę (mało zbiorów na dyskietce i konieczność zmiany jej stron). Pozytywną cechą nowego modelu jest także pełne 512 KB pamięci RAM i nowoczesna, zmieniona obudowa. Dyskusyjne może być natomiast zastosowanie czarno-białego monitora, takiego samego jak w modelu PC 1512.

Uwzględniając potrzeby użytkowników dodano wyjście równoległe CENTRONICS, które umożliwia podłączenie drugiej drukarki lub plottera. Zapomniano niestety o złączu RS 232C istotnym przy korzystaniu z modemu. Nie przewidziano też dołączenia twardego dysku do komputera, co utrudnia realizację dużych baz danych. Zmieniono klawiaturę, nadając jej typowy dla IBM wygląd.

Modyfikacje oprogramowania dotyczą głównie Locoscripta i produktów z nim współpracujących. Oprócz nowej wersji edytora dostarczany jest również Locospell i Locomail, programy służące do badania poprawności tekstu i ułatwieniu pracy korespondencyjnej. Do CP/M-u dodano sterowniki programowe (drivery) drukarek EPSON FX 80 i DIABLO 630 (daisywheel).

Reasumując, zmieniono wygląd zewnętrzny komputera, poprawiono niektóre rozwiązania sprzętowe, ale wybór drukarki rozetkowej spowodował przesunięcie głównego akcentu na operacje tekstowe. Dlatego sądzę, że z punktu widzenia naszego rynku najważniejszy jest fakt, że wprowadzenie nowego modelu obniżyło cenę starszych typów

Dane techniczne:

Procesor:	Zilog Z80, 4 MHZ
Dodatkowe procesory:	Intel 8048, 8041
RAM:	512KB
ROM:	0 KB (króciutki program ładujący w układzie scalonym obsługującym drukarkę)
Pamięć zewnętrzna	stacja dysków 720 KB 3 [illegible]
opcjonalnie:	druga taka stacja
Ekran:	90 kolumn, 32 wiersze (720×256 punktów) czarno-biały
wejście/wyjście:	równoległe wyjście typu Centronics, wyjście na drukarkę AMSTRADA
Klawiatura:	82 klawisze o układzie dopasowanym do Locoscripa
System operacyjny:	CP/M Plus
Drukarka:	rozetkowa o szybkości 20 znaków na sekundę, papier 15 cali
Oprogramowanie:	Locoscript, BASIC Mallarda, GSX, DR LOGO

Zalety:
1. Kompletny zestaw do pracy
2. Duży ekran (40% więcej znaków niż na typowym)
3. Stacja dysków o dużej pojemności
4. Klawiatura podobna do IBM PC
5. Drukarka o wysokiej jakości pisma
6. Niska cena
7. Brak pamięci ROM, możliwość ingerencji w działanie sprzętu na drodze czysto programowej
8. Zwarta i estetyczna konstrukcja
9. Dedykowany edytor tekstów
10. Popularny system operacyjny CP/M Plus

Wady:
1. Brak złącza RS 232 C w typowym zestawie
2. Wolna i głośna drukarka
3. Trudność [illegible] na drukarce
4. Brak grafik[illegible]

Jonasz Mayer

SHOOT'EM-UP to w wolnym tłumaczeniu strzelanina a jednocześnie najpopularniejszy typ gier komputerowych należących do wielkiej rodziny gier ARCADY. Strzelaj do wszystkiego, co się rusza lub wygląda groźnie — to podstawowa zasada tego typu gier. EXOLON jest właśnie najlepszym tego przykładem.

Firma HEWSON znana jest z produkcji prawdziwych hitów gier komputerowych np. URIDIUM, GUNRUNNER (z gatunku ARCADY), AVALON, DRAGONTORC, ASTROCLONE (z gatunku ADVENTURE). Właśnie EXOLON (produkt firmy z połowy '87 roku) okupował przez dłuższy czas pierwsze miejsca na listach gier komputerowych. Otrzymał też liczne nagrody m.in. CLASIC (Sinclair User), CRASH SMASH (Crash) za grafikę czy tzw. „playabitity" — możliwość wyżycia się w grze.

Celem gry jest niszczenie wszystkiego, co masz na swojej drodze. Musisz niszczyć wrogów ruchomych jak i przeszkody napotkane na drodze (możesz też trafiać w pociski nieruchomych transporterów stojących na drodze). Musisz też byc bardzo ostrożny (zdalnie kierowana rakieta nie jest do zestrzelenia — zestrzel sterownik rakiety), liczy się też refleks — bez niego daleko nie zajdziesz. Na twojej drodze roi się też od min, które wybuchają jak tylko na nie nadepniesz. W zaznaczonym miejscu na mapie możesz wymienić broń (dostajesz też swego rodzaju tarczę przeciw minom).

Wskazówki!

- aby strzelać rakietami musisz przytrzymać dłużej „Fire"
- skacz bardzo precyzyjnie
- pionowa bariera siłowa znika po dłuższym strzelaniu w nią z broni ręcznej
- bądz ostrożny przy nieruchomych transporterach
- nie staraj się zestrzelić wszystkiego, to może się źle dla ciebie skończyc.



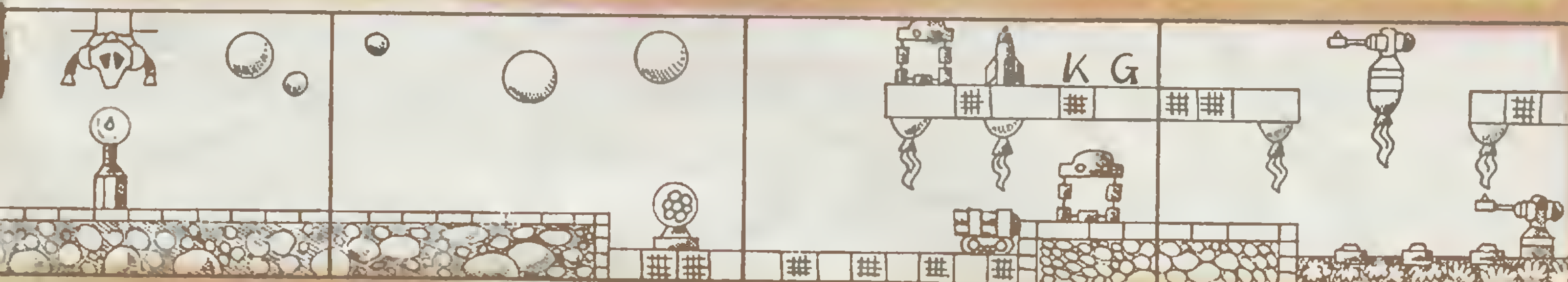
Nie potrafię powiedzieć, ile stref trzeba przejść do konca, ale ja przeszedłem ich 123 i strefa nr 101 jest powtórką pierwszej a następne kombinacjami poprzednich.

Na koniec tzw. „CHEET" czyli oszustwo: wybierz definiowanie klawiszy i wpisz po kolei litery Z,O,R,B,A. Usłyszysz dźwięk i po ponownym zdefiniowaniu klawiszy (lub wybraniu drążka) będziesz „nieśmiertelny".

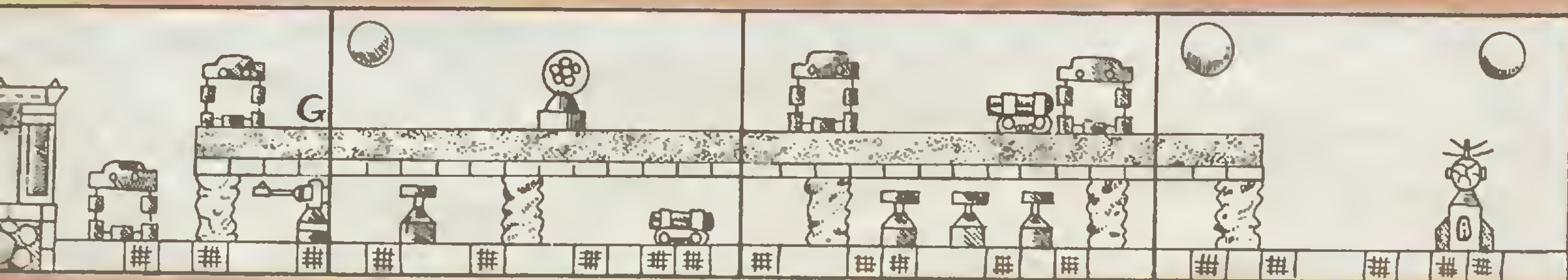
Wesołej zabawy!

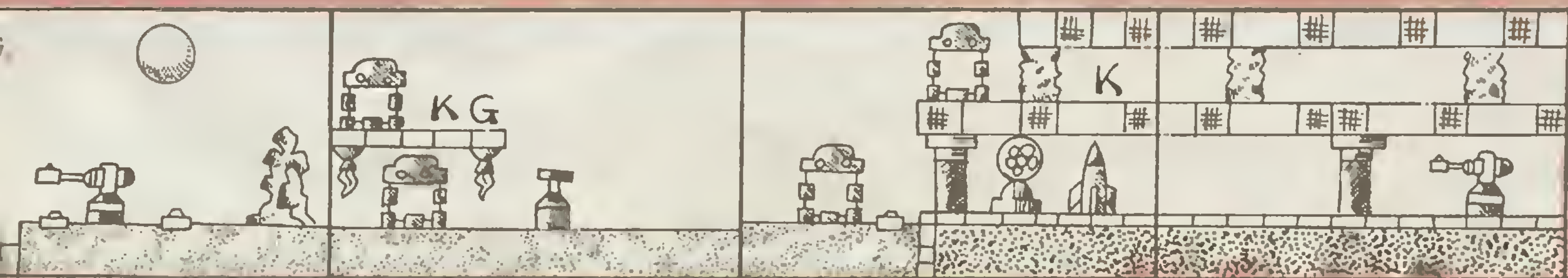
M.1

Autor: RAFFAELLE CECCO
Firma: HEWSON CONSULTANTS
Komputer: ZX Spectrum 48/+, Commodore 64/128, Amstrad/Schneider.



ⓄRYS: STAWICKIⓄ

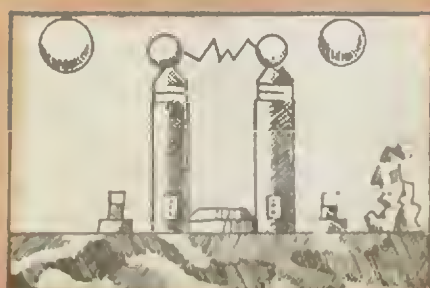






MAGAZYNEK DO KARABINU
—— " —— GRANATNIKA

PODNOSNIK
LEPIEJ TO ZNISZCZ BO ZGINIESZ
TO TEZ WARTO ROZWALIC
UWAGA !!! MINA



 KOSZYCZEK Z KULKAMI - ROZWAL (

- DZIAŁKO

↓ - TU WYMIENISZ BROŃ

 TELEPORT

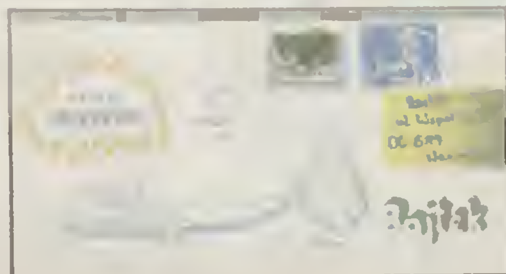
 - STEROWANE RAKIETY - ZNISZCZ

10 BAJTKOWA LISTA PRZEBOJÓW (3/88)

I znów lawina listów z typami najlepszych dziesiątek. Gusta Czytelników są bardzo zróżnicowane, lecz koncentrują się wokół gier ostatnio opisywanych w Bajtku. Nowe pozycje Listy staramy się jak najszybciej opatrywać opisem. I dlatego proszę, aby — w miarę możliwości — wraz z propozycją Dziesiątki nadsyłać opisy gier nowych, byśmy mogli zaprezentować je Czytelnikom. Na kopercie proszę o wyraźny dopisek „OPIS" obok „Lista Przebojów". Na obecne notowanie nadeszło 3171 propozycji na 137 tytułów

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	DAM BUSTERS	↑		×	×	×
2	SABOTEUR II	!		×	×	×
3	EQUINOX	↑		×	×	×
4	POP EYE	↓	×	×	×	×
5	ANTIRIAD	↓		×	×	×
6	BARBARIAN	!	×	×	×	×
7	ARKANOID	!	×	×	×	×
8	MIKIE	!		×	×	×
9	MISS PACMAN	↓	×	×	×	×
10	GLADIATOR	↓		×	×	×

Nagrody otrzymują: Justyna Pacak i Dominik Sokołowski oboje z Warszawy.



Kopertę nadesłał Norbert Launikjaniec z Bielska-Białej.



MIKIE

Każdy z was zapewne marzy o łobuzowaniu w szkole. Gra MIKIE pozwala to robić prawie bezkarnie. Jeżeli jesteś na tyle zręczny, by wydostać się ze szkoły, by pójść na randkę ze swoją dziewczyną, gra ta nie będzie dla ciebie trudna.

Zbierając serca, które leżą w każdym pomieszczeniu układasz pewien napis — liścik od ukochanej. Po ułożeniu listu biegnij do drzwi z napisem OUT — wyjdziesz na szkolny korytarz. Tam czeka na ciebie woźny ze szczotką i nauczyciel biegnący za tobą. Idź do drzwi z napisem IN i wciśnij przycisk FIRE. Znalazłeś się w kuchni. Jeżeli jesteś w niebezpieczeństwie podejdź do sterty brudnych talerzy i rzuć jednym z nich w nauczyciela. Zacznie on się otrzepywać, ty wtedy uciekniesz.

Następną komnatą jest stołówka. Na pewno dasz sobie radę, pomocne Ci będą leżące na stołach udźce baranie. Gdy rzucisz nim w goniącego Cię nauczyciela, on zamiast gonić Cię, zacznie obgryzać udziec.

Na sali gimnastycznej unikaj ćwiczących uczniów. Ćwiczą oni karate i gdy zbliżysz się do któregoś, zostaniesz powalony na ziemię. Uratować Cię może tylko szybkie podniesienie się z parkietu i ucieczka przed nadbiegającym profesorem.

Uffff! Te drzwi prowadzą na dziedziniec szkolny. Lecz tu znowu przeszkoda — trzech woźnych goniących naszego bohatera. Widok dziewczyny przed bramą na pewno Cię zmobilizuje i podbiegniesz do niej wymijając woźnych. Potem tylko szybki pocałunek i, trzymając się za ręce, biegniecie do kina.

Pamiętaj o tym, że profesor, gdy zdenerwuje się pogonią, zatrzymuje się i rzuca swą sztuczną szczęką. Jest ona bardzo groźna, gdyż natychmiast odsyła Cię do szpitala.

Gdybyś nie mógł dosięgnąć któregoś z serc, krzyknij głośno w jego stronę — serce spadnie.

Doskonała grafika i jeszcze lepsza muzyka (notabene jest to „A Hard Day's Night" zespołu The Beatles) to niewątpliwe atuty tej gry. Grając w MIKIE na pewno niczego się nie nauczysz, ale przynajmniej rozerwiesz.

Autor: Jonathan Smifff
Firma: Imagine/Konami
Komputer: ZX Spectrum 48/+, Commodore 64/128, Amstrad/Schneider.

(mateo)

KRÓL I KRÓLOWA GIER



JUSTYNA PACAK lat 12, chodzi do szóstej klasy Szkoły Podstawowej Nr 85 w Warszawie. W domu ma Spectrum, najbardziej lubi grać w ORTOGRAFIE. Interesuje się biologią, lubi zwierzęta. Ma żółwia i rybki. Lubi też roboty ręczne i rysowanie. Plany na przyszłość wiąże z ...



DOMINIK SOKOŁOWSKI lat 15, chodzi do I klasy LO im. S. Batorego w Warszawie. Posiada ZX Spectrum. Jego ulubiona gra to SABOTEUR II. Interesuje się informatyką, sportem i muzyką komputerową. Swe plany na przyszłość wiąże z komputerami. W szkole uczy się LOGO na lekcjach informatyki.



ANTICS

Na pewno wiele razy chwaliłeś się, że jesteś bardzo odważny, wytrzymały, że umiesz sobie poradzić w każdej sytuacji. Jeśli chcesz przekonać się, czy tak rzeczywiście jest, sięgnij po grę ANTICS. Co prawda nie będziesz miał okazji wykazać się swą siłą, ale sprawdzisz swą zręczność, pamięć i... cierpliwość.

Czy widzisz tę pszczołę na ekranie, w pobliżu ula? Wystarczy odrobina fantazji i już wyobrażasz sobie, że to Ty nią jesteś. Niestety, nie będziesz latał beztrosko po łące w poszukiwaniu najpiękniejszych kwiatów (tak jak w The Birds and The Bees — pierwszej części ANTICS). Spotkało Cię nieszczęście — Twój brat Borys został porwany i uwięziony w wielkim podziemnym mrowisku. Jeśli nikt nie przyjdzie mu z pomocą, Borys zginie — z dala od słońca i kwiatów. Czy mógłbyś na to pozwolić? Z drugiej strony, wiesz przecież, że aby dotrzeć do niego, trzeba pokonać wiele niebezpieczeństw, narażać własne życie.

Czego się nie robi dla brata? Wkraczasz więc w tajemniczy świat mrówek. Na pewno przyda Ci się zamieszczona obok mapka — przestudiuj ją uważnie. Oto kilka informac[illegible] zapewne przydatnych.

Mrówki wydrążyły w ziemi ogromny labirynt z mnóstwem ślepych korytarzy, zamaskowanych przejść, które będą otwierać się pr[illegible]ed Tobą po dotknięciu odpowiedniego kw[illegible]tka — włącznika. Czy zauważyłeś krążąc[illegible] bezustannie mrówki i inne owady? To m[illegible]ańcy podziemnego mrow[illegible]. Mrówki [illegible] swego siedliska i będą [illegible]bowały prz[illegible]odzić [illegible] w wyprawie. [illegible] nimi odbiera Ci częsc [illegible] pot[illegible]ebnych do [illegible]cia i latania. Strata [illegible] — [illegible]tkie nadzieje na uwolnienie b[illegible] będ[illegible] str[illegible]

[illegible] przez labirynt [illegible]gląd[illegible] uważn[illegible] w poszukiw[illegible] kwiatów. [illegible] ich tam n[illegible]wiele, ale są dla [illegible] wię[illegible] zauważysz kwiat, szybko usiądź na nim. Od razu poczujesz się silniejszy a na skrzydełkach osiądzie Ci dużo pyłku. Nie obawiaj się latających kul — są tylko po to, by Cię nastraszyć. Również małe gąsiennice biegające tu i ówdzie są całkowicie nieszkodliwe. Dość niebezpieczne są biedronki — dotknięcie jej odbiera część siły, ale biedronka zatrzymuje się na chwilę i pozwala Ci przejść.

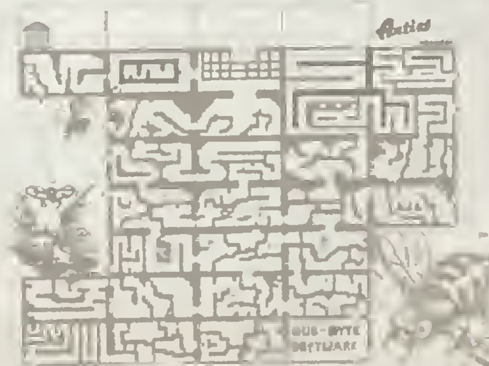
Pamiętaj, by nie siadać na małym, niepozornym kopczyku — to pułapka.

Podczas gry będziesz informowany, ile masz pyłku (Pollen), siły (Stamina) oraz jaką część zadania wykonałeś. Czy wiesz, dlaczego mrówki porwały Borysa? Posiadł on ich sekret — tajemnicę niezniszczalności zakodowaną w rosnących w mrowisku kwiatkach. Musisz je więc wszystkie zebrać, aby móc wrócić z bratem do ula.

Przez cały czas będzie towarzyszyła Ci muzyczka ale nie daj się zwieść jej przyjemnym tonom. Miej się ciągle na baczność. Borys czeka na Ciebie, musisz go uwolnić.

Firma: Bug-Byte Software
Komputer: ZX Spectrum 48/+, Amstrad/ /Schneider.

(aw)



S.O.S.

Bardzo proszę o nieśmiertelności do gier: Pyjamarama, Enigma Force, Pentagram. Nie wiem, jak skończyć gry: Under Wurlde, Dun Darach, Uboat Hunt, Hobbit. O co chodzi w grach: Jet Man, Empire, Green Beret, Reversi, Backgammon?

Maciej Samcik
ul. Przybyszewskiego 66/13
60-357 Poznań

Mam komputer Spectrum. Poszukuję następujących gier: Aliens, Rambo, Cobra Stallone, Winter Games, Gladiator, Jack The Nipper, Panzadrome. W zamian 15 innych.

Tomasz Oleksa
ul. Olchowa 5/61
20-355 Lublin

Mieszkam w Libii. Mam komputer ZX Spectrum +, ale znudziły mi się wszystkie gry. Może ktoś przyśle mi coś ciekawego na ten komputer. Mam też problem z uruchomieniem gry Marsport. I jeszcze jedna prośba — proszę o opis gry Miami Vice.

Piotr Benza
Benghazi — Libia
PO Box 8396

Proszę o dokładny opis gry Biggles PT II na Commodore C64

Maciej Tralewski
ul. Baligrodzka 3/15
59-300 Lublin

Mam duże kłopoty w przejściu niektórych gier, więc zwracam się o pomoc: jak przejść z pierwszego do drugiego etapu i dalej w grach Quasimodo, Droids, Choplifter w wersji na Atari 800 XL. I druga prośba: jak korzystać z programu muzycznego Musiccompos także na Atari?

Paweł Orzechowski
ul. Orkana 18c/28
34-410 Rabka

Poszukuję dokładnego opisu języka MEGA BASIC na Spectrum i opisów gier na ten komputer.

Jarosław Musiatik
ul. Pawia 37/40
59-300 Lublin

Uprzejmie proszę o opis gry ATIC-ATAC na ZX Spectrum

Michał Dyakowski
9 Orpheus Road
Reddiff
ZIMBABWE, AFRYKA

Pomocy! Nie wiem, jak zacząć i jak grać w grę Broad Sides oraz Winter Olympic. Nie wiem także, jak bawić się grą Bruce Lee i jak uzyskać nieśmiertelność w grze Chuckie Egg. Mam Atari 65 XE i... bez waszej pomocy nie dam sobie rady

Hanna Kucewicz
ul. Kosciuszki 8/3
14-520 Pieniężno
woj. elbląskie

Jestem posiadaczem Atari 65 XE. Potrzebuję pomocy w grach: Chimera i Zeppelin.

Dawid Pastwikowski
ul. Czarnieckiego 35/16
14-100 Ostróda

Mam wielki kłopot. Nie wiem, jak uzyskać nieśmiertelność w grach na Amstrad CPC 464. Nie wiem także, jak bawić się grami: Premier, Fruit Machine, Sultan's Maze, Flight Path 737. Proszę także o instrukcję do gier: Sorcery, The Survivor, Mutant Monty. Liczę na waszą pomoc.

Tomasz Trygar
al. Pokoju 10/39
31-548 Kraków

Kto udzieli mi informacji dotyczącej gry „V" na Commodore 64? Nie potrafię jej uruchomić, tzn. nie mogę się wydostać z pomieszczenia przy jakimś samolocie. Co mam zrobić?

Daniel Mękal
ul. Szczepowa 7a
71-750 Szczecin

Proszę o pomoc w następujących grach: Zorro, Road Race, Pitfall II, Speed King, Quest For Tires (po przeszkodzie z żółwiami i ptakiem) w wersji na Atari 65 XE.

Robert Olszewski
ul. Żuromińska 3/114
03-341 Warszawa

Mój komputer to Spectrum +. Proszę o pomoc w grach: Under Wurdle, Atic Atac, Monty Mole. Jak uzyskać nieśmiertelności w grach: Krakatoa, Commando, Cassey Jones?

Michał Sobczak
ul. 3-go Maja 17/12
62-800 Kalisz

Poszukuję dokładnego opisu do g[illegible] [illegible]rro, Raid Over Moscow, Pole Position w wersji na Ata[illegible] XL

Dorota Cybula
Oracze
19-325 Straduny
woj. suwalskie

[illegible] was [illegible] mam [illegible] że dzięki S.O.S. znaj[illegible] [illegible] gry Bru[illegible] [illegible]sz bardzo dokładny [illegible] Mo[illegible] [illegible] Atari [illegible] XE [illegible] w[illegible]

Damian Babicz
ul. Pocztowa 5/39
16-100 Sokołka

[illegible]

Tomasz Kruszyna
ul. Nowy Świat 12/5
28-400 Pińczów

KOMPUTER'88

KOLOROWY ZAWRÓT GŁOWY



W ciągu czterech dni przez sale PKiN w Warszawie, gdzie swoją siedzibę miała wystawa KOMPUTER przewinęły się nieprzebrane tłumy. Mimo jednak, że na zapleczu każdego stoiska bez przerwy toczyły się poufne negocjacje handlowe, do informatycznego Edenu równie daleko nam w marcu jak i przed wystawą.

Teoretycznie na KOMPUTER' 88 nie było łatwo się dostać. Tylko przez dwie godziny dziennie wystawa była otwarta dla zwiedzających. Jednak organizator wystawy AGPOL nie okazał się konsekwentny. W zasadzie każda firma mogła zaprosić kogo chciała. W efekcie żaden chyba ze strażników PKIN nie potrafił odróżnić jednego ze 150 rodzajów zaproszeń (tyle było wystawców) od pierwszego lepszego świstka. Dominującym doznaniem podczas całej wystawy była w tej sytuacji totalna ciasnota. Dla jej przezwyciężenia zdesperowani wystawcy zagraniczni proponowali przeniesienie wystawy do Poznania.

Ilość wystawców była rekordowa. Niestety jednak większość „wielkiego komputerowego światka" reprezentowana była przez pośredników, dystrybutorów, dealerów. W efekcie chociaż na stoiskach stały naprawdę najnowsze maszyny, nie było specjalnie z kim o nich porozmawiać.

W dziedzinie sprzętu jak przystało na wystawę profesjonalistów, dominowały „klony" IBM PC XT/AT. Nie zabrakło m.in. na stoisku IBM najnowszej rodziny amerykańskiego superkoncernu — rodziny PS2. Amstrad pokazał m.in. PCW 9512 a wyemancypowany Schneider PC 2640 świetnie prezentującą się maszynę kompatybilną z AT, lecz o nowej 3,5 — calowej już stacji dyskietek. Wystawiła ją w swej ofercie firma „Universal" z Berlina Zachodniego. Spośród peryferii uwagę przyciągały drukarki. Ich ceny spadły tak znacznie, że wielu drobnych pośredników pójdzie chyba z torbami. Mowa oczywiście o cenach w dolarach i ofercie wielkich firm — „Stara" i „Citizena". Pierwszy pokazał LC-10 (piszemy o niej w wywiadzie), drugi — drukarkę 120 D, której test zamieścimy w następnym numerze „Bajtka". „Star" i „Citizen" zamierzają, jak się wydaje, stworzyć wraz z kilkoma innymi firmami tamę przed zalewem naszego rynku dalekowschodnią tandetą.

Z softwarem na KOMPUTERZE 88 było chyba gorzej niż z „żelastwem".

Zagraniczni producencl, oprócz firmy „Soft-tronik", nie pokazeli nic ciekawego. Na ekranach monitorów systemów oferowanych naszym klientom przeważały zegary i obrazki przedstawiające, nia wiadomo dokładnie czemu, niezbyt sympatycznego pawiana. Jakby dla podkreślenia, że nic z prezentowanych maszyn na miejscu wycisnąć się nie da, w wielu stoiskach klawiatury były poodłączana, a zaproszonych przecież na wystawę gości oddzielały od sprzętu kolorowe łańcuchy. Ludzie, i to ma być prezentacja ofarty handlowej? — chciałoby się zapytać.

Oferta software'owa ze strony firm polskich też nie była zbyt bogata. Nic dziwnego, że nieliczne dobre programy mimo horandalnych cen były kupowane na miejscu. Dużym powodzeniam cieszył się m.in. zintegrowany pakiet technicznego przygotowania produkcji prazantowany przez „Interams" i systemy CAD/CAM która oferowało „Hapeko". W dalszym ciągu nie było natomiast żadnego np. edytora tekstu z prawdziwego zdarzenia. Może akorka niewarta wyprawki, bo i tak szybko ktoś to skopiuja.





Programowo-sprzętowym przebojam wystawy były oczywiście sieci. W ofarcia każdej firmy przeczytać można było o wielodostępie. Pojawiły się takża liczne modemy, także posiadające już polski atast. Do prawdziwych systemów brakuja jeszcze tylko, bagatelka, sieci telefonicznej z prawdziwego zdarzenia.

O handlu i planiądzach myśleli podczas wystawy niamal wszyscy oglądający, mówiła o tym część, a jeszcza mniej podejmowało jakieś rozmowy, niektóra z nich powinny dać interesujące efekty. Mówiło się podczas targów o nawiązaniu współpracy pomiędzy zakładami MERA w Błoniu, a którymś z ważniejszych w świecie producentow drukarak. Myśli się poważnia o powołaniu joint vanture.

Niektore firmy zagraniczne wiedząc, że polski klient nie śmierdzi dewizami zaproponował transakcje kompensacyjne. Pojawią się zatem na naszym rynku komputery kupowana za cukier, cebulę i włoszczyznę.

Na wystawę przyjechało tylko kilku poważniejszych biznesmenow. To oni właśnie marzyli o „joint venturas", o wielkim przerobie i średnim nawet zysku. Na polskim rynku komputerowym — powiedział Norbert Wieschalla, szef reprazentującej „Citizena" firmy „Synelec" — cena produktu i jego wartość to różne rzeczy. Na razie trudno o naprawdę dużego kontrahenta — a tylko takiego szukamy. Przy naprawdę dużym biznesie 1 proc. marży też da nam zarobić. Tylko czy znajdzie się partner do takiego interesu?



Organizacja wystawy nie była wprawdzia najlepsza, ala przyjemnie było chodzić między stoiskami nie tylko ze względu na migające obrazy na monitorach. Kilka firm do obsługi swoich stoisk poprosiło dziewczyny o oszałamiającej urodzie. Na dodatek konkurowały one w dziedzinie krótkości spódniczak. Taka właśnie dekoracja podobała się „Bajtkowi" najbardziej, chociaż jak stwierdził jeden z nas panie były w pełni kompatybilne lecz niemerytoryczne.

Wystawa była nie tylko prezentacją ofart. W trakcie jej trwania zorganizowano cykl konfarencji dla profesjonalnych użytkowników PC. Ich uczestnicy omawiali m.in. desktop publishing, edukację informatyczna, grafikę. Poziom poszczególnych spotkań i referatów był zróżnicowany. Nie wiadomo jeszcza co znaczy byc w naazych warunkach profasjonalistą.

Dla hobbystow KOMPUTER 88 zdecydowania nie był rajem. Brakowało w ogole komputerów 8-bitowych, mało było taż do nich programów. Cykl oprogramowania edukacyjnego wystawiła tylko firma „Intarsoft". Trudno byłoby zatem, może poza wspomnianymi drukarkami i ofertą „Atari" polecić cokolwiek czytalnikom „Bajka". Popatrzeć natomiaat było miło. Ci, którzy zwiedzili wystawę z pewnością zapamiętają jej obraz. Będzie on stawał jak żywy przed oczyma, gdy popatrzy aię na pusta półki w salonach „Unitry".

Grzegorz Onichimowski







NIE MA PROGRAMÓW DOBRYCH

... są tylko programy, w których na razie jeszcze nie znaleziono błędu. Takie powiedzenie dość często można usłyszeć wśród ludzi zajmujących się oprogramowaniem komputerów. Spróbujmy zastanowić się, co to, na pozór dziwne, stwierdzenie może oznaczać dla programistów, a co dla użytkowników maszyn cyfrowych.

Zacznijmy tworzyć duży program. Początkiem takiego działania musi być zadanie, które ten program ma rozwiązywać. Jeśli jestem informatykiem hobbystą, to zwykle zadanie stawiam sobie sam. Jeśli mam programowaniem zarobić na życie, to muszę rozwiązywać zadania postawione przez innych — przyszłych użytkowników mojego oprogramowania. Analizuję zadanie, tworzę koncepcję rozwiązania, przyglądam się jej, jeśli nie jest zadowalająca szukam lepszej. Ostatecznie powstaje jakiś, mniej lub bardziej szczegółowy projekt, według którego następnie można pisać program. Potem przychodzi radosny moment napisania ostatniej instrukcji. Jako człowiek rozsądny, przed dalszymi eksperymentami zapisuję program w bezpiecznym miejscu, np. na dyskietce, jako człowiek doświadczony przez życie robię na drugiej dyskietce zapasową kopię tego zapisu i mogę powiedzieć: gotowe. Zwykle jednak radość jest krótkotrwała.

Mało prawdopodobne jest, że pisząc kilkaset linii programu nie pomylimy się ani razu. Toteż zwykle pierwsze próby uruchomienia naszego dzieła kończą się wypisaniem przez maszynę komunikatu o znalezieniu błędnych konstrukcji języka programowania. A to przecinek zamiast kropki, a to znowu BGIN zamiast BEGIN. Zdarzają się również dużo bardziej wyrafinowane, a co za tym idzie trudniejsze do zauważenia, błędy składniowe, ale to trochę inny temat. Dla naszych dzisiejszych rozważań istotne jest, że wszystkie te błędy wykryje translator języka programowania i tym samym wymusi na nas ich usunięcie — bez tego nie da się wykonać programu. Zresztą dla osoby dobrze znającej język, w którym programuje, poprawienie tych błędów nie jest dużym problemem.

PROGRAM ZACZYNA DZIAŁAĆ

Ostatecznie, po kilku próbach i poprawkach nasz program wykonuje się od początku do końca i produkuje jakieś wyniki. Podkreślam JAKIEŚ, bo często zdarza się, że nie są to wcale takie wyniki, o jakich marzyliśmy. Może np. zdarzyć się, że odległość z miasta A do miasta B, obliczona przez program wynosi — 5 km, liczba murarzy potrzebna do zbudowania domu przez tydzień wyszła równa 1,48, a program, który miał być super wróżką, przewidział koniec świata na rok 1823. Wszystkie te wyniki mają jedną wspólną cechę: są mianowicie, delikatnie mówiąc, mało wiarygodne. Cecha ta skłania zwykle autora do wnikliwego przyjrzenia się programowi i wprowadzenia w nim pewnych zmian (najbardziej skrajnym przypadkiem takich zmian jest wyrzucenie całości i rozpoczęcie prac od punktu wyjścia).

Zwykle jednak i ten etap zostaje szczęśliwie pokonany. Program daje wyniki, które wyglądają przyzwoicie i chyba są dobre. Owszem, mogą być dobre, ale wcale nie muszą. W programie dalej mogą być błędy! Choćby tak! zamiast A=1000 jest A=10000, głupi drobny błąd, a wynik jest dziesięć razy za duży! Wniosek jest jeden: nawet wyników, które „na oko" są porządne, nie możemy od razu zaakceptować, trzeba je sprawdzić. Jak? Musimy wykonać program dla takich danych, dla których znamy poprawne wyniki i porównać rezultaty pracy programu z tym, o czym na pewno wiemy, że jest dobre. Często oznacza to, że nasze zadanie musimy sami rozwiązać „ręcznie", aby uzyskać odpowiedni materiał do prowadzenia testów programu. I tutaj zaczyna się równocześnie kilka wątków. Zajmijmy się nimi kolejno.

Jeśli okaże się, że program działa źle, zwykle stajemy przed poważnym problemem: dlaczego? Lokalizacja błędów tego typu jest często bardzo trudna, ta problematyka odbiega jednak od naszych dzisiejszych rozważań, zajmiemy się nią kiedy indziej.

Jeśli program jest pisany na zlecenie, to pomoc zleceniodawcy może być niezwykle przydatna przy sprawdzaniu poprawności. Np. może on dostarczyć danych do testów, z gotowymi poprawnymi wynikami. Zauważmy również, że jako przyszły użytkownik programu, zleceniodawca powinien (w swoim własnym interesie) dołożyć wszelkich starań aby upewnić się, że program który ma otrzymać działa poprawnie. Oczywiście szanujący się informatyk będzie starał się sam wykryć jak najwięcej błędów, żeby potem nie najeść się wstydu.

TRZY RAZY NIE

No i wreszcie rzecz najważniejsza: powiedzmy, że przetestowaliśmy nasz program na dziesięciu różnych zestawach danych i zawsze otrzymaliśmy właściwy wynik. Czy wiemy już, że nasz program jest w pełni poprawny? NIE, NIE i jeszcze raz NIE. Podchodząc do problemu ściśle (a takie podejście jest tu niezbędne) możemy stwierdzić tylko jedno: program działa poprawnie dla tych dziesięciu zestawów danych. Jakie będą wyniki dla innych danych — tego nie możemy być pewni. Oczywiście, wiemy że program zbudowany jest zgodnie z logiką, więc jeśli liczy dobrze dla jednych danych, to tak samo powinno być dla innych. Na tej podstawie domyślamy się, wnioskujemy, mamy nadzieję,

jesteśmy głęboko przekonani, że cały program jest poprawny, że dla wszystkich danych da dobre wyniki. Ale pewności mieć nie możemy, chyba że przetestujemy program na wszystkich możliwych zestawach danych, co jest zwykle nie do wykonania w praktyce*), albo udowodnimy, że program jest zawsze poprawny, tak jak w szkole dowodziliśmy np. twierdzenia Pitagorasa. Tego zaś zwykle nie jesteśmy w stanie zrobić, bo mimo ogromnych wysiłków badaczy skierowanych na opracowanie skutecznych metod dowodzenia poprawności programów, praktyczne efekty są jak dotąd niewielkie.

Czy jednak cały ten problem nie jest trochę wydumany? Napisałem poprzednio „ściśle biorąc nie mamy zupełnej pewności". Takie sformułowania na milę pachną teoretyzowaniem. Na pewno nie, problemu poprawności, czy też niezawodności oprogramowania nie wymyślili teoretycy, choć, jak już wspomniałem, wielu z nich biedzi się teraz nad jego rozwiązaniem. Problem ten zauważyli praktycy, a może raczej odczuli go na własnej skórze i to często bardzo boleśnie. Oczywiście nie dotyczy on na ogół programów małych, mieszczących się w kilku liniach, lecz większych, liczących setki czy tysiące instrukcji. Jednak prosty, przykład, który zaraz podam może się przemknąć także w niedużym programie:

```
readln (a,b); {czyta wartości zmiennych a i b}
writeln (1/(a-b));
```

te instrukcje mogą się wykonywać poprawnie nawet całe lata, dopóki liczby wczytywane jako dane nie będą sobie równe. Wtedy koniec — próba dzielenia przez zero, program pada. Życie dostarcza zwykle dużo bardziej skomplikowanych przykładów konstrukcji, które są poprawne prawie zawsze, tylko w pewnych, szczególnych przypadkach dają bezsensowne wyniki. Co gorsza, często wcale nie powodują awarii programu — liczy się on do końca, tylko wynik jest bez sensu.

GDZIE SZUKAĆ RATUNKU?

Programy trzeba testować, testować wszechstronnie, metodycznie i oczywiście inteligentnie. Starać się rozpatrzeć wszystkie możliwe przypadki, szczególnie te złośliwe (jak w naszym przykładzie — gdy obie liczby równe), szczególnie te graniczne. Np. co będzie jak podam długość równą zero, albo baza danych — co się stanie jak usunę wszystkie rekordy — czy nie ulegnie dezorganizacji, itd. Należy też dokładnie zbadać odporność programu na błędy w danych — czy je wykrywa i właściwie sygnalizuje. Bardzo dobre efekty daje zwykle przetestowanie programu nie przez twórcą, lecz przez kogoś innego.

Testowanie musi być tym dokładniejsze i wszechstronne im więcej zależy od poprawnego działania programu. To nie żarty — np. od programu sterującego ruchem pociągów zależy życie setek ludzi.

Powiecie, że takie dokładne i wszechstronne testowanie pochłonie więcej czasu niż samo programowanie. I oczywiście macie rację! Według danych pochodzących z firm produkujących oprogramowanie, więcej niż połowa kosztów tegoż oprogramowania, to właśnie koszt testowania i poprawiania gotowych programów.

Ustaliliśmy, że programista powinien zrobić wszystko, aby jego programy były wolne od błędów i że może się zdarzyć, że to nie wystarczy i błędy jednak wystąpią. Co w tej sytuacji powinien robić użytkownik oprogramowania?

Musi upewnić się, że rzeczywiście program był testowany, warto także przetestować go samodzielnie jeszcze raz. Choćby dlatego, że programista mógł pewne szczegóły zadania zrozumieć nieco inaczej niż zleceniodawca i program działa bezbłędnie, tylko... rozwiązuje nieco inne zadanie niż miał rozwiązywać. Nigdy i nikomu nie wolno bezkrytycznie akceptować danych tylko dlatego, że wyliczył je komputer.

I wreszcie, każdy kto korzysta z komputera musi pamiętać, że nie ma programów dobrych, za dobre uważa się te programy, w których na razie nie znaleziono błędów.

Andrzej Pilaszek

---

* Często możliwych zestawów danych jest nieskończenie wiele, poza tym, gdybyśmy znali poprawne rozwiązania dla wszystkich danych, to program nie byłby potrzebny.

CENY NA DZIEŃ 07.02.1988

	GIEŁDA „BAJTKA" (tys. zł)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
SINCLAIR			
ZX 81	30	—	39
ZX Spectrum 48 KB	100	115	110–150
ZX Spectrum Plus	140	—	180–230
ZX Spectrum 128 + 2	240	—	250
Drukarka SEIKOSHA GP 50S	90	—	99
TIMEX 2048	140	146	—
Joystick	4,5–7	—	4–6
COMMODORE			
C-64	220	219	320
C-128	320	299	480
C-128D	—	—	999
Amiga 500	1,4 mln	—	2800
Magnetofon 1531	45	48	30
Stacja dyskietek 1541	200	—	399
Stacja dyskietek 1571	240	299	460
Drukarka GP-500	190	—	149
Dyskietki 5 1/4 (średnia jakość)	0,8–1,5	3,5	0,3–1,5
ATARI			
65 XE	140	125	110
130 XE	210	199	320
Stacja dyskietek 1050	230	187	350
ATARI 520 STM st. dysk. 0,5Mb	1,1 mln	798	800
AMSTRAD			
464 z monit. monochromat.	280	—	400
6128 z monit. monochromat.	450	—	750
6128 z monitorem kolorowym	550	—	1000
Dyskietki 3	5	—	6–9
Stacja dyskietek 3 do 464	380	—	399
PC 1512 SD MD	1,0 mln	—	470

Giełdy Bajtka odbywają się przy ul. Grzybowskiej 35 (Szkoła Podstawowa nr 25) w każdą sobotę i niedzielę od 14.00 do 19.00

CZKAWKA

Ceny sprzętu na giełdzie wreszcie drgnęły z dużą szansą na dalszy ich spadek. Być może jest to efekt obniżki o 200 dolarów ceny Pewexowskiej mikrokomputera Atari 520 lub zapowiedzi Centralnej Składnicy Harcerskiej wprowadzenia do sprzedaży Spectravideo X16 oraz przenośnego mikrokomputera Bondwell.

Nowości sprzętowych nadal niewiele, a największa wystawa „Komputer'88" także nie zaprezentowała rewelacji. Uczniowie wprawdzie ciągle czekają na „Juniora — 800", ale jeżeli Elwro zachowa dotychczasowe tempo produkcji słynny „Junior" stanie się przestarzałym sprzętem nawet dla seniorów. Wprawdzie „Unimor" opracował kolejny model ośmiobitowego mikrokomputera, — Bosman oparty na podzespołach polskich i z krajów socjalistycznych lecz niestety wsad dewizowy także będzie, a zatem jeśli nawet mowi się o skali produkcji i cenie, czy warto tego słuchać?

No cóż, uczniowie! Odwiedzajcie giełdę „Bajtka". Tam przynajmniej na pewno kupicie upragniony sprzęt. Przy okazji informacja dedykowana raczej odpowiedzialnym za rozwój mikrokomputeryzacji dla młodzieży. W Moskwie otwarto kolejne przedszkole z klasą komputerową, gdzie dzieci uczą się posługiwać komputerem. Specjalnie dla nich zmieniono nawet budowę klawiatury i rozkład barw monitora. *(gr)*



INDYWIDUALNY BANK DANYCH

Radosław Dubella, lat 15. Posiada mikrokomputer C-64, magnetofon, dwa joysticki, około 1500 programów i gier. Proponuje wymianę doświadczeń oraz gier.
Adres: 75-347 Koszalin, ul. Władysława IV 58a/37.

Marcin Pośpiech, uczeń lat 11. Posiada ZX Spectrum 48 KB oraz magnetofon Panasonic. Zainteresowania: informatyka, sport i plastyka. Oprogramowanie: gry i programy użytkowe. Nawiąże kontakt w sprawie wymiany informacji o grach, proponuje wymianę gier.
Adres: 53-404 Wrocław, ul. Stalowowolska 4/7.

Krzysztof Pawłowski, uczeń lat 15. Posiada Atari 800 XL oraz około 200 programów. Proponuje wymianę doświadczeń i programów.
Adres: 98-300 Wieluń, ul. Kopernika 13.

Wiesław Albrant, pragnie nawiązać kontakt listowny z posiadaczami Atari 800 XL lub 65 XE w celu wymiany programów a głównie doświadczeń w dziedzinie zastosowania małych mikrokomputerów do półprofesjonalnych zadań w projektowaniu, obliczaniu konstrukcji.
Adres: 01-926 Warszawa, ul. Kwitnąca 8 m 13.

Justyna Łaszewska, lat 10. Uczennica Szkoły Muzycznej I-go stopnia. Posiada Commodore+4, joystick, magnetofon 1531 i telewizor. Proponuje wymianę gier i programów użytkowych
Adres: 81-704 Sopot, ul. Kościuszki 9 m 2.

Jarosław Żarczyński, uczeń lat 17. Posiada Commodore VC-20, magnetofon 1531, telewizor w systemie Pal-Secam. Programy: użytkowe — matematyka, fizyka, gramatyka, oraz gry.
Adres: 11-200 Bartoszyce, ul. Nad Łyną 9/16.

Krzysztof Kraska, lat 14. Komputer: Commodore C-128, monitor 1902A, stacja dysków, drukarka. Oprogramowanie: programy muzyczne, użytkowe oraz sporo gier. Zainteresowania informatyka i elektronika. Proponuje wymianę programów literatury oraz doświadczeń (szczególnie programy graficzne na drukarkę).
Adres: 66-400 Gorzów Wlkp. ul. Walczaka 15 a/1.

Irena Kiedroń, studentka lat 20. Posiada Atari 800 XL, magnetofon XC 12. Nawiąże korespondencję w celu wymiany literatury, doświadczeń i oprogramowania. Korespondencja w języku polskim.
Adres: 735-64 Havirov-Sucha, CSSR, ul. Srazarmovska 13.

Tomasz Karaśkiewicz, uczeń LO, 17 lat. Posiada komputer Amstrad-Schneider 464. Oprogramowanie: programy użytkowe, edukacyjne oraz gry. Proponuje wymianę programów oraz doświadczeń.
Adres: 99-300 Kutno, ul. Zamoyskiego 9/64.

Ziemowit Ogrodowski, uczeń 14 lat. Posiada mikrokomputer C 16 + 64 kb, magnetofon 1531. Oprogramowanie: około 100 programów. Proponuje wymianę programów z użytkownikami C-16 z powiększoną pamięcią do 64 kb.
Adres: 63-500 Ostrzeszów, ul. Bolesława Śm...ego 5.

Marek Adamus, inżynier mechanik. Posiada mikrokomputer Atari 800 XL, magnetofon XC 12. Zainteresowania: radio-elektronika. Oprogramowanie: około 100 gier oraz programów użytkowych. Proponuje wymianę oprogramowania, literatury i doświadczeń.
Adres: 41-908 Bytom, ul. Nowa 19a/6.

SPRZĘŻENIE ZWROTNE

Drogi Bajtku!

Marcin Waligórski odpowiada na listy czytelników.

Zwracamy się z uprzejmą prośbą o poinformowanie nas, w jakiej wersji TURBO-PASCAL-a zrealizowanej dla komputera AMSTRAD CPC 6128 możliwe jest używanie takich rozkazów jak:
— overlay,
— window,
— draw,
itp.
gdyż w posiadanej przez nas wersji 3.0 te rozkazy traktowane są jako błędne.

Z podanej przez Was informacji (Bajtek nr 7/87) wynika, że jest możliwe nakładkowanie programu w TURBO PASCAL-u.

O ile taka wersja istnieje, prosimy o wskazanie instytucji, która umożliwiłaby sprzedaż w/w kompilatora.

mgr inż. Michał Hała
MULTIREAKTOR PL
Oddział w Gliwicach

Turbo Pascal v. 3.0 jest najbogatszą wersją języka Turbo Pascal, dostępną w systemie CP/M Plus. Do niedawna była to także najbogatsza wersja tego języka dla komputerów klasy IBM PC — obecnie wypiera ją nowy Turbo Pascal v. 4.0.

Wersja 3.0 w żadnej ze swoich implementacji nie jest wyposażona w standardowe procedury graficzne, w tym Draw, Window itp. Procedury graficzne firma Borland dostarcza w postaci bibliotek procedur w Pascal-u, które dołącza się do pisanych przez siebie programów. W przypadku systemu operacyjnego PC-DOS przykładem (i to dobrym) takiego pakietu procedur może być **TURBO GRAPHIX TOOLBOX**. Według posiadanych przeze mnie informacji żaden jego odpowiednik nie został przez firmę Borland wydany dla komputera Amstrad 6128. Sprawą tą zajęły się natomiast inne firmy — i w rezultacie powstało kilka tego typu produktów o różnych możliwościach. Przykładem może być pakiet procedur wydany przez niemiecką firmę Heimsoeth. Niestety, redakcja „Bajtka" nie dysponuje informacją, kto w Polsce rozprowadza to oprogramowanie.

Opcja nakładkowania procedur przez użycie słowa kluczowego **overlay** została opisana w podręczniku „Turbo Pascal Reference Manual" bez żadnych zastrzeżeń dotyczących pracy w systemie CP/M. Jak jest w rzeczywistości — wskazuje praktyka. Programista nie pozostaje jednak bez wyjścia. Można bowiem jeszcze kompilować części programu na różne pliki — a właściwie podzielić duży program na kilka mniejszych, a następnie „nakładki" skompilować przy ustalonej opcji CHN-FILE. Taka kompilacja powoduje utworzenie pliku wynikowego bez dołączenia doń biblioteki procedur standardowych. Program taki może być następnie z poziomu programu „głównego" wczytany do pamięci i wykonany przy pomocy procedury **Chain**. Jej parametrem jest nazwa „odpowiedniej zmiennej plikowej". Oczywiście program „główny" kompilujemy w zwykły sposób (COM-FILE), gdyż podczas wywołania **Chain** w pamięci musi się znajdować biblioteka procedur standardowych.

Trzeba pamiętać, że wywołanie programu przez **Chain** to nie to samo, co użycie procedury — nie następuje bowiem powrót do miejsca wywołania! Z tego powodu przy użyciu **Chain** możemy wykonać jedynie łańcuch kolejnych programów, z których każdy wywołuje swojego następcę (stąd nazwa „chain files").

Problemem pozostaje przekazywanie parametrów do wywołanych programów. Można to robić różnymi sposobami. Nie wykraczając poza standard Pascal-a — poprzez zapis parametrów na osobny plik dyskowy. Jest to jednak rozwiązanie opóźniające wykonanie programu. Można więc np. przekazywać wartości parametrów poprzez określone miejsce w pamięci, np. zarezerwowawszy je uprzednio przez odpowiednią modyfikację parametrów kompilacji START ADRESS i END ADRESS. Można też robić to przy pomocy zmiennych globalnych programu, deklarując je według zasad odpowiadających deklaracji COMMON w FORTRAN-ie. Szczegóły znaleźć można w książce J. Bieleckiego „Turbo Pascal 3.0", WNT 1987.

Jest nas trzech początkujących brydżystów. Niestety nie mamy „czwartego" i z tego powodu nie możemy uczyć się gry praktycznej. A nikt, kto dobrze gra, nie będzie tracił czasu na grę z nami. Inna rzecz, że nikogo takiego nie znamy. Programy brydżowe, o których wiem (na C-64) są przeznaczone tylko dla jednej osoby. W związku z tym mam dwa pytania:

1. Czy jest taki program brydżowy na C-64, który zastępuje czwartego gracza?

2. Czy jest możliwość przeróbki normalnego programu brydżowego tak, aby symulował on (w jednej z opcji) tylko jednego gracza?

Bogusław Rdzanek
Warszawa

Redakcji znane są 4 takie programy.
— LEARNING BRIDGE uczy chyba wszystkiego, co z grą w brydża jest związane (z wyjątkiem tasowania kart i, oczywiście, oszukiwania). Niestety, program dostępny jest tylko w wersji dyskowej.
— BRIDGE 2.0,
— BRIDGE 4.0 — ulepszona wersja BRIDGE 2.0, gra nieco rozsądniej,
— CONTRACT BRIDGE.
Red. Klaudiusz Dybowski zapewnia, że powyższe programy (bez BRIDGE 4.0) można uzyskać w ursynowskim klubie MANIAK.

Jestem posiadaczem mikrokomputera ZX Spectrum +. Interesuje mnie, gdzie w ROM-ie zapisany jest program obsługi Basicowych komend SAVE i LOAD?

Witek
(nazwisko i adres do wiad. redakcji)

Zazwyczaj w opisach systemu operacyjnego ZX Spectrum podaje się dwie procedury obsługujące transmisję na linii komputer — magnetofon. Są to:

SAVE — BYTES 04C2. Procedura służy do nagrywania ciągu bajtów na taśmę. Wymagane parametry: w rejestrze IX — adres początku nagrywanego bloku, w DE — jego długość, w A — typ nagrywanego bloku (0 dla nagłówka pliku, FF dla pliku właściwego).

LOAD — BYTES 0556. Procedura odczytująca ciąg sygnałów z wejścia EAR. Wymagane parametry: w rejestrze IX — adres, od którego ładowany blok ma się rozpoczynać w pamięci, w A — jego typ (0 lub FF, ewentualnie inne wartości, jeśli wczytujemy pliki o nadanych przez nas typach). Dodatkowym parametrem jest wartość bitu C (Carry). Dla C = 0 wykonywana jest weryfikacja danych w bloku, dla C = 1 — ładowanie.

Wejście do wnętrza tych procedur pozwala niekiedy uzyskać efekty specjalne w postaci np. ruchomych pasków na obrzeżu ekranu. Przykładem może być tu często spotykana w różnych spisach efektów specjalnych instrukacja RANDOMIZE USR 1333, 1331, 1290, 1269, 1251, 1367 itd.

Polecam zapoznanie się z dostępną literaturą:
K. Kuryłowicz, D. Madej, K. Marasek „Przewodnik po ZX Spectrum", wyd. I, WKŁ 1986, rozdz. 4.7.1 i 4.5.3 oraz A. Kadłof „Tajniki ZX Spectrum", wyd. KWCz i miesięcznik „Komputer", rozdz. VIII.

Czy emulator to rozwiązanie hardware'owe, czy programowe?
(nazwisko i adres do wiad. redakcji)

Takie, albo takie, zależnie od sposobu wykonania i przeznaczenia. Spotykamy np. programowy amulator systemu CP/M na komputerze Atari 520 ST, oraz np. sprzętowy emulator komputera IBM PC dla komputera Amiga 1000. Ogólnie rzecz biorąc, w celu emulacji programów stosuje się raczej rozwiązania programowe zaś w celu emulacji konfiguracji sprzętowych — raczej rozwiązania sprzętowe. Jest jednak sporo wyjątków od tych reguł.

Interesuje mnie dziedzina informatyki związana z przetwarzaniem obrazu naturalnego (chodzi o użycie komputera Atari 520 ST w celu produkcji efektów specjalnych dla potrzeb filmów video). Proszę o informację, czy dostępna jest jakakolwiek literatura w języku polskim na ten temat? Jeżeli tak, to proszę chociby ogólnikową informację o zakresie podejmowanych w niej zagadnien.

Tomasz Rubinowicz
Kraków

Spieszę z odpowiedzią, bo jest jeszcze nadzieja, że w chwili ukazania się lutowego numeru „Bajtka", w jakiejś zapomnianej przez wszystkich księgarni dostanie Pan potrzebną Panu książkę. Jest nią właśnie wydana przez WNT w serii Biblioteki Inżynierii Oprogramowania pozycja Theo Pavlidis „Grafika i przetwarzanie obrazów", wyd. I, WNT 1987. Nakład 7 tys. egzemplarzy gwarantuje, że książka już bardzo niedługo będzie nie do zdobycia.

Praca ta obejmuje szereg zagadnień związanych zarówno z przetwarzaniem obrazu naturalnego, jak też z syntetyzowaniem obrazu z użyciem algorytmów geometrii trójwymiarowej. Oto niektóre z poruszanych problemów:

a. Obróbka obrazu naturalnego poprzez korektę jasności, kontrastu, barw, filtrowanie, analizę kształtu;

b. Algorytmy używane w grafice trójwymiarowej w tym kolorowanie, zasłanianie;

c. Generowanie obrazów trójwymiarowych na podstawie danych geometrycznych, w tym tworzenie obrazów krzywych, powierzchni i brył.

Niestety, jest to jedyna publikacja, jaką mogę Panu bez wahania polecić. Jak zapewne zdaje sobie Pan sprawę, przetwarzanie obrazów, zwłaszcza dla potrzeb techniki filmowej, jest w naszym kraju bardzo mało popularną dziedziną zastosowań informatyki — z oczywistą szkodą dla rozwoju tej techniki.

Marcin

Szanowny Panie!
Wiadomości dotyczące przeróbki klawiatury ZX Spectrum podane w Pana liście wykorzystałem. W załączeniu przesyłam zdjęcie obrazujące wyniki przebudowy.
Klawiatura działa poprawnie, na życzenie służę materiałami informacyjnymi.

Łączę pozdrowienia
J. Ludwiński
Radom



WARTO PRZECZYTAĆ

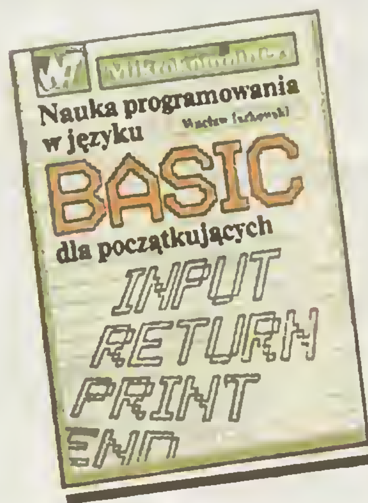


Wydawnictwa Naukowo-Techniczne sprawiły miłą niespodziankę czytelnikom pasjonującym się informatyką. W znanej serii „Mikrokomputery" ukazało się ostatnio kilka książek. Duże nakłady gwarantują, że w najbliższym czasie nie będziemy musieli ich poszukiwać na giełdach i w antykwariatach.

Wszystkim, którzy od niedawna mają do czynienia z komputerem lub dopiero zamierzają się nim zainteresować proponuję na początek książkę ***„Basic dla wszystkich"****, której autorami są* ***JACQUES BOISGONTIER i SOPHIE BREBION.*** *Gwarantuję, że będzie ona bardzo użyteczną lekturą. Autorzy kierowali swoje opracowanie głównie do użytkowników TRS-80 i Apple II, lecz większość zawartych w niej informacji przyda się również przy korzystaniu ze Spectrum, Atari czy Commodore lub też Amstrada. Podstawowe instrukcje i zasady składni Basica są prawie identyczne w każdej jego implementacji.*

Książka jest napisana w bardzo przystępny sposób. Każdy rozdział zawiera króciutkie podsumowanie, w którym znajdziemy najistotniejsze wiadomości. Książka ma duże walory dydaktyczne, może pełnić rolę elementarza Basicu dla początkujących.

Drugą pozycją zajmującą się wyłącznie Basic-em jest ***„Nauka programowania w języku Basic dla początkujących" WACŁAWA ISZKOWSKIEGO.*** *Porusza ona znacznie trudniejsze problemy niż „Basic dla wszystkich". Korzystanie z niej bez przygotowania nie będzie takie proste, pominąwszy może pierwsze rozdziały wyjaśniające pojęcia algorytmu, programu i jego struktury oraz przedstawiające architekturę komputera. Dalsza część skierowana jest do osób mających już pewne przygotowanie. „Basic dla początkujących" nie jest więc samouczkiem, zawiera usystematyzowany wykład Basica, w którym stosowana jest fachowa terminologia. Książka ta może być bardzo przydatna jako pomoc w prowadzeniu zajęć na kursach*



i w klubach mikrokomputerowych, gdzie zapewniona jest opieka doświadczonego wykładowcy.

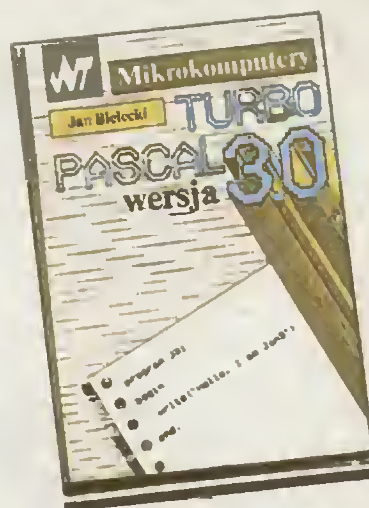
Autor nie ograniczył się do opisania instrukcji i składni Basica. Omawia oprócz tego na licznych przykładach prawidłowy styl programowania. Podawanym wiadomościom towarzyszą przykłady oraz rysunki. Bardzo dobrym pomysłem jest załączanie do każdego rozdziału słowniczka angielskich terminów.

Obie książki ułatwiają ogólne poznanie Basica, lecz przystępując do pracy z konkretnym komputerem powinniśmy zapoznać się dokładniej z zaimplementowaną w nim wersją. Najistotniejsze różnice występują zwykle w przypadku instrukcji graficznych, i dźwiękowych. Także sposób organizacji pamięci może być powodem nieporozumień. Radziłbym więc uzupełnić biblioteczkę książkami zajmującymi się wyłącznie naszym komputerem np. opisany w numerze 1/88 „Bajtka" ***„Commodore-Basic" KLAUDIUSZA DYBOWSKIEGO,*** *czy* ***„Przewodnik po ZX Spectrum" K. KURYŁOWICZA, D. MADEJA I K. MARASKA.***

Miłą wiadomością dla wszystkich, którzy zetknęli się ze Spectrum, Meritum, czy Amstradem będzie ukazanie się książki ***„Mikroprocesor Z 80" JERZEGO KARCZMARCZUKA.*** *Poznanie języka wewnętrznego komputera i architektury mikroprocesora, to już wyższy stopień wtajemniczenia. W pierwszej części proponowanej książki znajdziemy wyczerpujący opis budowy i organizacji mikroprocesora Z-80, jego działania oraz szczegółowe omówienie rozkazów. Druga część zapoznaje nas z techniką programowania w asemblerze. Zrozumienie poruszanych tu problemów nie jest łatwe, ale włożony w to wysiłek opłaci się. Dzięki temu nie tylko lepiej poznamy nasz komputer, ale również zdobędziemy nad nim nieograniczoną władzę. Sprawne posługiwanie się asemblerem umożliwi poprawienie niedoskonałego „Basica", napisanie własnych ciekawych procedur, a także efektywne korzystanie z systemu operacyjnego.*



Jako lekturę uzupełniającą do „Mikroprocesora Z 80" proponowałbym książkę ***„Mikroprocesor w pytaniach i odpowiedziach" A. SACHY,*** *która daje okazję do zapoznania się z nowoczesną techniką mikroprocesorową.*



Ostatnią książką o tematyce dotyczącej języków programowania jest ***„Turbo Pascal wersja 3.0" JANA BIELECKIEGO.*** *Zawiera ona wyczerpujący materiał opisujący Turbo Pascal, a więc dyrektywy umożliwiające korzystanie z systemu, omówienie składni instrukcji, procedury, typy itp. Autor posługuje się dużą liczbą trudnych pojęć informatycznych bez komentarza, który umożliwiłby ich zrozumienie. Książka przeznaczona jest więc raczej dla studentów niż amatorów samodzielnej nauki Pascala.*

Przydałoby się również kilka krótkich, przykładowych programów praktycznie ilustrujących sposób zastosowania instrukcji PASCAL-a.

(jj)

Jacques Boisgontier, Sophie Brebion „Basic dla wszystkich", przekład Anna Iszkowska, Wydawnictwa Naukowo-Techniczne, Warszawa 1987. Wyd. I. Nakład 80.000 egz. cena 380 zł.
Wacław Iszkowski **„Nauka programowania w języku BASIC dla początkujących"** Wydawnictwa Naukowo-Techniczne, Warszawa 1987, Wyd. I,. Nakład 100.000 egz., cena 450 zł.
Jerzy Karczmarczuk **„Mikroprocesor Z 80",** Wydawnictwa Naukowo-Techniczne, Warszawa 1987, Wyd. I,. Nakł. 30.000 egz., cena 400 zł.
Jan Bielecki **„Turbo Pascal wersja 3.0",** Wydawnictwa Naukowo-Techniczne, Warszawa 1987, Wyd. I,. Nakł. 40.000 egz., cena 590 zł.

SAMI O SOBIE

OPEN SOSNOWIEC

Klub nasz powstał w maju 1987 r. Jest więc klubem rozwijającym dopiero swoją działalność, a jednocześnie mogącym się pochwalić pewnymi osiągnięciami. Sponsorem i opiekunem klubu jest Sosnowiecka Spółdzielnia Mieszkaniowa. Posiadamy mikrokomputery ATARI 800XL, 130XE, stacje dysków, drukarkę 1029, monitory kolorowe, oprogramowanie w postaci programów użytkowych i gier (preferując gry logiczne).

Klub jest otwarty dla wszystkich, którzy pragną poznać świat mikrokomputerów, rozwijać swoje zainteresowania informatyką, lub skorzystać z literatury. Oferujemy pomoc tym, którzy mają problemy w zakupie sprzętu, jego eksploatacji lub przy tworzeniu swoich własnych programów. Praca odbywa się w grupach specjalizujących się w grafice, dźwięku i grach logicznych.

Organizujemy również pokazy możliwości posiadanego sprzętu i jego praktycznego wykorzystania dla szkół z terenu miasta jak i mniejszych miejscowości. Aktualnie rozpoczynamy drugi kurs obsługi i programowania mikrokomputerów ATARI. Dla najmłodszych klub oferuje w każdą środę tygodnia pakiet gier. Obecnie staramy się nawiązać współpracę z Wydziałem Oświaty Urzędu Miejskiego w zakresie prowadzenia zajęć z elementów informatyki dla szkół wykorzystując nasz sprzęt, jak również szkolenia nauczycieli mających prowadzić zajęcia z tego przedmiotu. Pragniemy w najbliższym czasie utworzyć sekcję zajmującą się sprawami hardware'u.

Zapraszamy wszystkich chętnych do współpracy z nami tak z kraju jak i z zagranicy. Popieramy propozycję klubu MR ATARI z Kozienic stworzenia centrum informacji komputerowej i jesteśmy chętni doprowadzić wspólnie tę propozycję do finału.

Zapraszamy chętnych do wymiany oprogramowania jak i doświadczeń, których na pewno nazbierało się dużo w trakcie dotychczasowej działalności.

Nasz adres: Klub Mikrokomputerowy „OPEN", ul. ZMP 7 a 41—200 Sosnowiec.

Z poważaniem
Kier. Klubu Mikrokomputerowego „O P E N"
(R. Benc)

ATARI-OLSZTYN

Studio Mikrokomputerowe „Atari" powstało w lutym 1986 roku pod patronatem Spółdzielni Mieszkaniowej „Jaroty". Efektem ogłoszonych na łamach „Dziennika Pojezierza" zapisów do klubu było ponad dwa tysiące chętnych.

Początkowo Studio prowadziło przede wszystkim działalność pokazową, opartą oczywiście na mikrokomputerach ATARI. W następnej kolejności rozpoczęło organizowanie kursów programowania w językach „LOGO" i „BASIC". Klub nawiązał również współpracę z kilkoma olsztyńskimi placówkami oświatowymi, między innymi ze szkołą dla dzieci głuchoniemych a także szkołami podstawowymi nr 25 i 27. Nauczyciele i członkowie klubu twierdzą zgodnie, że współpraca układa się znakomicie, najbardziej jednak zadowoleni są uczniowie, którzy mogą dzięki temu zapoznać się z techniką komputerową i programami edukacyjnymi.

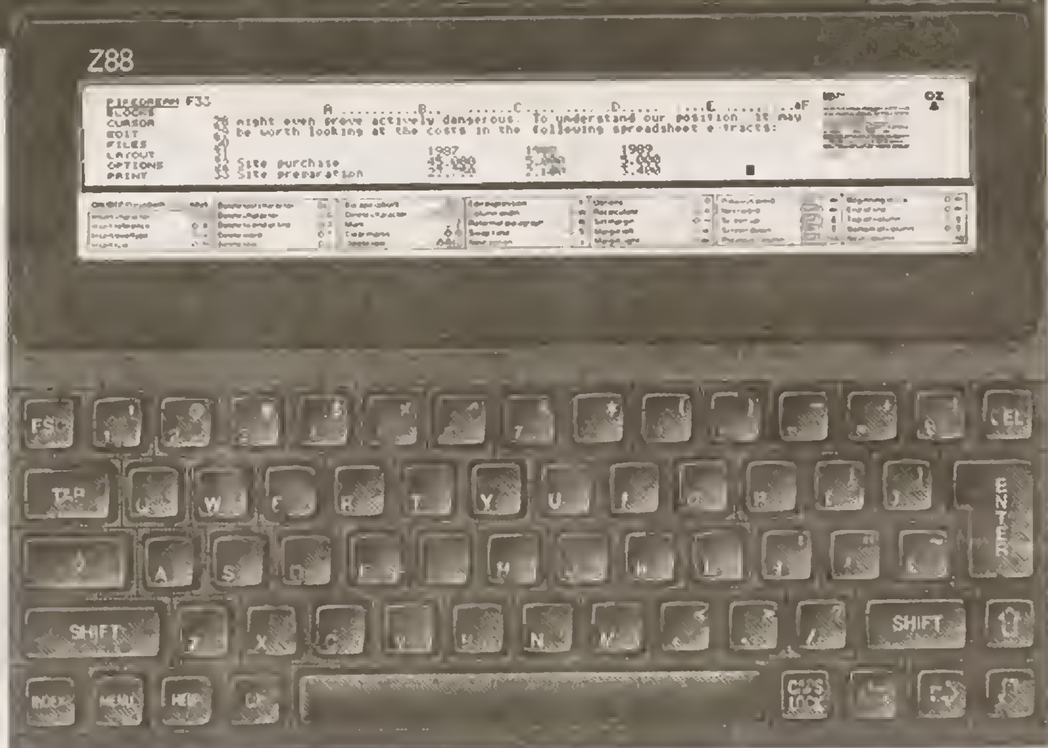
W kwietniu 1986 roku Studio, wspólnie ze Szkołą Muzyczną w Olsztynie zorganizowało w Filharmonii Olsztyńskiej imprezę pod nazwą „Muzyka z procesora". Zaprezentowano na niej muzyków korzystających w pracy z komputerów i — wówczas nowość techniczną — compact disc, gramofon laserowy.

Studio to jednak nie tylko pokazy, prezentacje, wystawy, giełdy i kursy programowania, to także zespół ludzi potrafiących pisać programy. Niektóre z ich mają charakter całkiem profesjonalny, np. program „Płace" napisany dla Spółdzielni Mieszkaniowej „Jaroty".

Adres do korespondencji:
Studio Mikrokomputerowe „Atari"
Olsztyn ul. Murzynowskiego 8

Ewa Staniszewska

CZY Z88 PODBIJE NASZE SERCA?



„Ma mniej niż cal grubości i wymiary formatu A4. Waży poniżej dwóch funtów. Zasilany jest czterema bateriami AA. I kosztuje £ 249.99 (+VAT). KOMPUTER PONAD WSZYSTKO".

Tak głosi slogan reklamujący nowy komputer Z88 Sir Clive'a Sinclaira. Spróbujmy zastanowić się, co kryje się w tym ostatnim zdaniu.

Z88 od początku projektowany był tak, aby mógł zapewnić właścicielowi wszystkie możliwości jakimi dysponuje „biurkowy" PC przy jednoczesnym zmniejszeniu masy i wymiarów. Dysponuje on ekranem ciekłokrystalicznym (LCD): 8 linii po 94 znaki. Ciekawostką jest możliwość korzystania z czterech „okien" oraz z „mapy" pokazującej, w którym miejscu tekstu aktualnie się znajdujemy. Klawiatura komputera wykonana jest z utwardzonej gumy, ale jest o wiele trwalsza i wygodniejsza w użyciu niż ta znana ze Spectrum.

Z88 posiada 32 K pamięci RAM, z czego dostępne jest ok. 15 K. Pamięć tę można rozszerzyć do 416 K za pomocą modułów 32 K (£20) i 128 K (£50). Przewidywana jest produkcja modułów 1 megabajtowych co pozwoli rozszerzyć pamięć do 3 M.

Jeżeli chodzi o pamięć zewnętrzną to zrezygnowano z dyskietek wprowadzając moduły EPROM (Erasable Programable ROM) o pojemnościach 32 K i 128 K (w przyszłości 1 Mega).

Komputer Z88 posiada port szeregowy RS 232 służący do wymiany danych z IBM PC (i kompatybilnymi) lub podłączenia dowolnej drukarki. Nie jest jednak kompatybilny z IBM PC ze względu na zastosowanie mikroprocesora Z80.

Dla wygody użytkownika wbudowano do ROM-u kilka najbardziej przydatnych programów użytkowych, takich jak: baza danych, procesor tekstu, kalendarz wraz z zegarem (+ alarm), notatnik. Większość funkcji wykonywanych przez te programy osiągalna jest przez naciśnięcie kombinacji kilku klawiszy (podobnie, jeżeli chodzi o „przeskakiwanie" z programu do programu).

Dostępne obecnie dodatkowe wyposażenie komputera to: zasilacz sieciowy, moduły RAM i EPROM, programy obsługujące port wejścia (wyjścia, kasownik EPROM, kabel RS 232. Przewidywana jest też produkcja modemu telefonicznego.

Czy komputer przyjmie się na rynku (przewidywana sprzedaż — 10.000 sztuk miesięcznie) — nie wiadomo. Do jego atutów należą: małe wymiary i niska cena, do wad — gumowa klawiatura i różne opinie na temat Sir Clive'a.

Dokładniejsze informacje na temat Z88 uzyskać można pod adresem:

The Z88 Information Desk
Dept. 1110 OB
Sidney House
Sussex Street
Cambridge
CB 1PA

na podst. The Observer, The Times

Michał Zeman

WSZYSTKO DLA WSZYSTKICH

ABC Data

FIRMA NA KTÓREJ MOŻNA POLEGAĆ

LISTA CEN

(Ceny eksportowe, bez MWSt, w markach RFN)

PRODUKTY STAR MICRONICS

DRUKARKI	Prędkość druku: Norm/NLQ	DM
LC-10	144/36	450
LC-10C (do C-64/128)	144/36	450
NX-15	120/30	750
ND-10	180/45	800
NR-10	240/60	1000
ND-15	240/60	1000
NR-15	240/60	1200
NB24-10 (24-igłowa)	216/72	1100
NB24-15 (24-igłowa)	216/72	1400
NB-15 (24-igłowa)	300/100	1980
AL-500 (komputer adresujący z drukarką	200/50	3800
Drukarki stosujące taśmę barwiącą na szpulkach:		
SG-15	120/30	900
DP-8340	110/—	550

NOWOŚĆ: *Od kwietnia oferujemy Państwu drukarkę* ***laserową Star Laser Printer 8****, oraz drukarkę* ***kolorową LC-10 colour!***

UWAGA: 1. *Drukarka SG-15 stosuje dostępną w Polsce taśmę na szpulkach oraz posiada 16kB pamięci wewnętrznej.*
2. *„10" lub „15" w nazwie drukarki oznaczają szerokość wałka w calach. DP-8340 ma wałek 5-calowy.*

Dodatkowe interfejsy:	**DM**
RS-232 do SG-15	120
16kB pamięć do drukarek N, NB (poza NL-10)	270
8kB RS-232 interfejs do N, NB (poza NL-10)	220
Kable podłączeniowe do komputerów:	**DM**
IBM PC/XT/AT lub kompatybilne; Atari ST	20
Amstrad 464, 664, 6128, Schneider 464, 664	20
Schneider 6128	20
Automatyczne podajniki kart:	**DM**
SF-10J (jedna komora; do LC-10)	160
SF-10D (jedna komora; do NL-10,ND-10,NR-10.NB24-10)	180
SF-15D (jedna komora, do NX-15, ND-15, NR-15)	370
SF-15B (jedna komora, do NB-15, NB24-15)	530
EB-2 (druga komora do SF-15B)	330

Kasety z różnym krojem trzcionki: do NB-15:	**DM**
Courier	120
Orator	120
Courier Italic	120
Prestige Italic	120
Letter Gothic	120
do NB24-15:	
Courier	120
Orator	120
Letter Gothic	120
Kaseta RAM (dla zdefiniowania dowolnych znaków)	190
Głowice do drukarek:	**DM**
Gemini-10X/10Xi,SG-10	150
Gemini-15X/15Xi,SG-15	150
Gemini-160/160i/160+s, Delta-10, SD-10	150
Delta-15, SD-15	170
Radix-10/15, SR-10/15	190
NL-10, LC-10	110
NX-15	130
ND-10/15	150
NR-10/15	340
NB24-10/15	480
NB-15	650
Taśmy barwiące:	**DM**
Kaseta do drukarek 10-calowych	21
Kaseta do drukarek 15-calowych	27
Taśma do kasety do drukarek 10-calowych	16
Taśma do kasety do drukarek 15-calowych	22
Szpulka z taśmą do Gemini-10X/15X/160, SG-10/15	9

Uwaga: *Zamawiając kasetę lub taśmę koniecznie prosimy podać typ drukarki.*

Urządzenie do regeneracji taśm:	**DM**
— na szpulkach	445
— w kasetach	395

star   Roland   DG   houston instrument   AMPEX

ABC Data Im-und Export GmbH
Augustastraße 40 5300 Bonn 2, RFN
tel. 0228/35.44.80,-90 telex 88.55.66

ZAMÓWIENIE

Niniejszym zamawiam następujące artykuły:

.... sztuk .... DM
.... sztuk .... DM
.... sztuk .... DM
.... sztuk .... DM
.... sztuk .... DM

Koszty transportu 1 sztuka DM 40,-
Kwota pobierana przez Dresdner Bank DM 10,-
Razem: DM ....

W załączeniu kopia zlecenia bankowego na przelew w/w sumy na konto ABC Data GmbH w Dresdner Bank 5300 Bonn 2, Numer konta: 268847500, Kod Bankowy (BLZ) 37080040

W/w sprzęt proszę wysyłać na adres:

Nazwisko i imię ....
Kod pocztowy i miejscowość ....
....
Ulica i numer domu ....
Telefon ....

Nazwisko, imię i adres zamawiającego — jeśli inne niż odbiorcy: ....
....

data .... podpis ....

ABC Data GmbH jest firmą zarejestrowaną w Amtsgericht Bonn, HRB 4058. Dyrektor: Lech Matusiak

WSZYSTKO DLA WSZYSTKICH

LISTA CEN

(Ceny eksportowe, bez MWSt w markach RFN)

Pokrywy do drukarek:	DM
Do modeli 10-calowych	20
Do modeli 15-calowych	30
Instrukcje serwisowe (do każdego modelu)	60

UWAGA. *Informujemy Państwa, że posiadamy pełen zestaw części zamiennych do drukarek Star*

PRODUKTY FIRMY ROLAND DG

PLOTERY		DM
DXY-880A	380x270 mm	1850
DXY-980A	380x270 mm	2250
DXY-885	416x276 mm	2450
DXY-990	416x276 mm	3050
DPX-2200	594x432 mm	8050
DPX-3300	864x594 mm	10270

PRODUKTY FIRMY HOUSTON

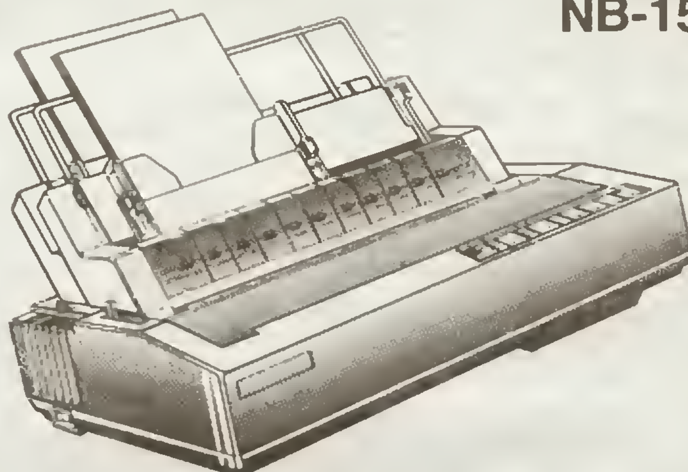
(UWAGA: *Ceny w dolarach amerykańskich*)

PLOTERY	Rozmiar	Dokładność (mm)	$
EDMP-29M	A3/A4	0,1	3240
EDMP-29M-3	A3/A4	0,1	4050
E595DN PC	A4	0,1	864
E795 DN PC	A4/A3	0,1	1290
EDMP-40	A4/A3	0,1	1830
EDMP-40-2	A4/A3	0,1	1830
EDMP-42	A2/A1	0,1	4392
EDMP-52	A2/A1	0,025	6324
EDMP-52-MP	A2/A1	0,025	7932
EDMP-55B	A4/A3/A2/A1	0,025	7140
EDMP-55B-MP	A4/A3/A2/A1	0,025	8460
EDMP-56B	A4/A3/A2/A1/A0	0,025	9780
EDMP-56B-MP	A4/A3/A2/A1/A0	0,025	11100
SCANNER			**$**
Scan-Cad Scanner Equipment Model 128			5250
DIGITIZER'Y			**$**
Digitizer'y serii 1000 (dokładność 0,381 mm, rozdzielczość 0,127 mm)			
ETG-1005 Digitizer	13x13 cm		510
ETG-1011 Digitizer	28x28 cm		906
ETH 1017 Digitizer	28x43 cm		1104
MT0580 Zasilacz 220 V			56
Kable podłączeniowe do komputerów (1 szt.)			65
Digitizer'y serii 8000 (dokładność 0,0254 mm, rozdzielczość 0,254 mm)			
TG-8011 Digitizer	28x28 cm		1122
TG-8017 Digitizer	28x43 cm		2238
TG-8024 Digitizer	46x61 cm		3954
TG-8036 Digitizer	61x91,4 cm		5274
MT0580 Zasilacz 220 V			56
Kable podłączeniowe do komputerów (1 szt.)			65

SYSTEMY KOMPUTEROWE KOMPATYBILNE Z IBM PC/XT/AT

	DM
PC-XT TURBO	
CPU 8088-2, 4,77/8MHz, 640kB RAM, licence BIOS, 2 X 360kB FDD, FDD/HDD controller, Hercules card parallel printer port, Multi I/O card, 135W or 150W power supply, Keyboard (84 keys), Metal case manual.	1490
PC-AT	
CPU 80286, 6/8MHz, 640kB RAM licence BIOS, 4 X serial 1 X parallel port on board, 1 X 1, 2 MB FDD, 1 X 360kB FDD, FDD/HDD controller, Hercules card parallel printer port, 200 W power supply, Keyboard (84 keys).	2750
PARCO MONITORS	**DM**
— Mono, 12", amber	250
— Mono, 14", amber	300
— Colour, 14"	
— Enhanced Colour, 14" (for EGA card)	1100
— EGA card	490
TERMINALS (AMPEX)	**DM**
— A-210 (głównie do tekstu)	760
— A-232 (tekst i grafika)	980

NB24-10
NB24-15
NB-15



JAK ZAMAWIAĆ ARTYKUŁY OFEROWANE PRZEZ ABC DATA?

1. **Drogą korespondencyjną:**

— dokonać wpłaty na nasze konto
ABC Data GmbH
Dresdner Bank, 5300 Bonn 2, RFN
Kod bankowy (BLZ): 37080040
Numer konta: 268847500

Prosimy tutaj pamiętać o doliczeniu kosztów transportu w wysokości DM 40 za każdą drukarkę lub ploter oraz o zaznaczeniu, że koszty bankowe związane z przelewem pokrywa wpłacający. Możecie Państwo również przesłać nam czek na odpowiednią sumę.

— po dokonaniu przelewu prosimy o wysłanie do nas załączonego zamówienia lub krótkiego listu najlepiej na odwrocie kserokopii dowodu wpłaty (z dokładną informacją o tym, co Państwo zamawiacie i na jaki adres towar ma być wysłany

2. **Osobiście:**

Wyroby nasze możecie Państwo również kupić osobiście w Hamburgu lub w Berlinie Zachodnim:

ABC Computer System GmbH
Wittenbergplatz 3a
1000 Berlin 30
tel. 213 59 37
Telex 183.00.0

ABC-Data GmbH
Ditmar-Koel-Str. 13
2000 Hamburg 11
tel. 31.40.03
Tx. 21.66.002

GWARANCJA

Na wszystkie drukarki STAR zakupione w firmie ABC Data udzielamy 1-rocznej gwarancji oraz zapewniamy serwis pogwarancyjny. Obsługa gwarancyjna, pogwarancyjna oraz informacja techniczna udzielane są przez:

Dom Handlowy Nauki Sp z o.o. PAN
00-950 Warszawa
ul. Filtrowa 83
tel. 659.52.11
telex 817 529
A. Malinowski, K. Jeziorski

Zakład serwisowy:
Copact Electronis
05-270 Marki k/Warszawy
ul. Świerczewskiego 72
tel. 19.32.35 w.143
K. Król, R. Ulewski

JAK Z NAMI NAWIĄZAĆ KONTAKT?

Najlepiej listownie, telegraficznie lub telexem. Możecie Państwo również do nas dzwonić — biuro nasze jest otwarte w godzinach 8.30-17.00. Poza tymi godzinami zgłosi się automat i będziecie Państwo mogli nagrać nam wiadomość. W takim wypadku prosimy o podanie nazwiska, adresu i krótkie wyjaśnienie sprawy, w której Państwo dzwonicie.

WARUNKI SPRZEDAŻY

CENY — należy je rozumieć jako z naszego składu w Hamburgu (FOB Hamburg).
— zastrzegamy sobie prawo zmiany cen.

DOSTĘPNOŚĆ — Jako **autoryzowany przez producenta** dystrybutor na Polskę, zapewniamy Państwa, że wszystkie oferowane przez nas drukarki STAR oraz plotery ROLAND i Houston są z zasady dostępne natychmiast z naszego magazynu w Hamburgu lub Berlinie Zachodnim.

TRANSPORT — za koszty transportu do Polski, odprawę celną oraz ubezpieczenie prosimy doliczyć DM 40 za każdą drukarkę, ploter lub komputer.

DOSTAWA — nasze transporty do Polski wysyłane są każdego tygodnia; gwarantujemy więc Państwu, że drukarki i plotery nadejdą w przeciągu 15 dni od otrzymania przez nas wpłaty. Okres dostawy sprzętu komputerowego jest 2-5 tygodni. Przesyłka dostarczana jest na podany adres, już po załatwieniu formalności celnych!

JUŻ PISZĘ!

Cześć Maluchy!

Propozycja, by nauczyć komputer pisać nie wydaje się być zbyt sensowną, przynajmniej na pierwszy rzut oka. Co jak co, ale pisać potrafi każdy, nawet najprostszy komputer. Po cóż więc go tego uczyć?

Mimo wszystko postanowiłem namówić Was, abyście Wasz komputer spróbowali nauczyć pisać i to przy pomocy żółwia LOGO. Domyślacie się, że „żółwiowe pisanie" będzie trochę inne niż „komputerowe", będzie mianowicie podobne do pisma odręcznego. Nasze literki będą musiały być niestety trochę kanciaste, aby żółwiowi łatwiej było je rysować.

Rozpocznijmy od literki m, jak mama. Najpierw spróbujmy ją narysować, najłatwiej będzie to zrobić na papierze w kratkę. Zwróćcie uwagę na to, że początek i koniec litery znajdują się na tej samej wysokości (dwóch kratek). Jest to konieczne, by kolejne litery łączyły się ze sobą.



Możemy się umówić, że kratka będzie miała boki o długości **x** a długość przekątnej oznaczymy przez **z**. Wytłumaczymy teraz żółwiowi jak ma rysować literę **m**

```
to m
 fd :z * 2
 rt 90 fd :z
 rt 45 fd :x * 3
 rt 180 fd :x * 3
 rt 45 fd :z
 rt 90 fd :z
 rt 45 fd :x * 3
 rt 180 fd :x * 3
 rt 45 fd :z
 rt 90 fd :z
 rt 45 fd :x * 2
 lt 45 fd :z
 lt 90 fd :z * 2
end
```

zanim jednak każdemu żółwiowi ją narysować, musimy mu powiedzieć ile wynosi **x** i **z**. Napiszmy więc na przykład:

?make „x 10
?make „z :x * 1.41

oznacza to, że zmienna **x** otrzymuje wartość 10 a zmienna **z** wartość zmiennej **x** pomnożoną przez 1.41. Dlaczego akurat 1.41? Dlatego, że przekątna kwadratu ma długość równą długości jego boku pomnożonej przez pierwiastek z 2 czyli w przybliżeniu 1.41. Kto nie wierzy, może to sprawdzić korzystając choćby z twierdzenia Pitagorasa.

Zaprojektujmy teraz literkę a:



i przetłumaczymy jej kształt na język LOGO

```
to a
 fd :z * 2
 rt 45
 repeat 12 [fd :x rt 30]
 pu
 fd :x * 2.5
 pd
 rt 90 fd :x * 3
 lt 45 fd :z
 lt 90 fd :z * 2
end
```

Tym razem trzeba było w pewnym momencie „oderwać pióro" od papieru. Zrobiliśmy to za pomocą rozkazu pu — podnieś pióro.

Znamy już więc wystarczająco dużo liter by napisać słowo mama. Napiszmy więc

? cs mama

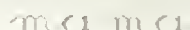
nie zapominając o przerwach (spacjach) pomiędzy literami. Na ekranie ukaże się nam następujący obrazek



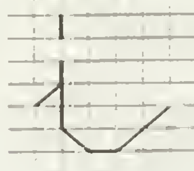
No tak, przecież żółw na początku ustawiony jest pionowo, nosem do góry ekranu. Żeby otrzymać poziomy napis trzeba żółwia obrócić w prawo o 45 stopni

? cs rt 45 mama

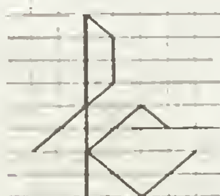
da w efekcie:



Rysowanie i pisywanie w LOGO kształtu innych liter nie sprawi już chyba nikomu specjalnych kłopotów. Dlatego podam tylko kilka przykładów. Resztę zrobicie sami.



```
to :
 fd :z lt 45 fd :x rt 180 fd :x * 3 lt 4
5 fd :z lt 45 fd :x lt 45 fd :z * 2
 pu lt 90 fd :z * 4 pd
 lt 135 fd :x
 pu lt 90 fd :x * 4 rt 90 fd :x * 3 lt 1
35
end
```



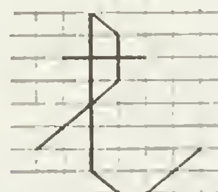
```
to k
 fd :z * 3 lt 45 fd :x * 2 lt 45 fd :z 1
t 135 fd :x * 8
 pu lt 135 fd :z * 3 pd
 lt 90 fd :z lt 90 fd :z * 2 lt 90 fd :z
 * 2 lt 90 fd :z * 2
end
```



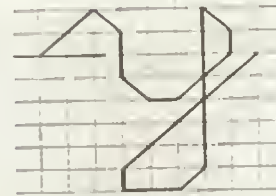
```
to o
 fd :z * 2 rt 45
 repeat 12 [fd :x rt 30]
 pu fd :x * 2.5 rt 90 fd :x * 2 pd
 lt 135
end
```



```
to r
 lt 45 fd :x rt 45 fd :z rt 45 fd :x rt
45 fd :z rt 45 fd :x * 3 rt 180 fd :x *
 3 rt 45 fd :z rt 135 fd :x lt 45 fd :z
 lt 45 fd :x lt 45
end
```



```
to t
 fd :z * 3 lt 45 fd :x * 2 lt 45 fd :z 1
t 135 fd :x * 7 lt 45 fd :z lt 45 fd :x
 lt 45 fd :z * 2
 pu lt 45 fd :x * 4 lt 90 fd :x * 2 pd
 fd :x * 3
 pu lt 90 fd :x * 4 lt 90 fd :x * 5 lt 4
5 pd
end
```



```
to y
 fd :z * 2 rt 90 fd :z rt 45 fd :x * 2 l
t 45 fd :z lt 45 fd :x lt 45 fd :z * 2
lt 45 fd :x lt 45 fd :z lt 135 fd :x *
7 rt 45 fd :z rt 45 fd :x * 2 rt 90 fd
:x rt 45 fd :z * 5
end
```

Warto jeszcze zdefiniować odstęp pomiędzy literami. Możemy oznaczyć go kropką

```
to .
 pu
 rt 45 fd :x * 4 lt 45
 pd
end
```

Wielkość liter zależy od wartości x i z. Dzięki temu możemy określić sobie różne rodzaje pisma.

```
to duze
 make "x 10
 make "z :x * 1.41
end

to male
 make "x 3
 make "z :x * 1.41
end

to srednie
 make "x 5
 make "z :x * 1.41
end
```

Początek napisu możemy ustalić sterując bezpośrednio żółwiem. Możemy również ułatwić sobie życie za pomocą takiej procedury:

```
to linia :nr
 pu home
 fd 190 lt 90
 fd 300 lt 90
 fd (:nr * 1.7 - 1) * :x * 8
 lt 135
 pd
end
```

Na zakończenie proponuję Wam wpisanie tego wiersza. Składa się on tylko z ośmiu liter, które już zdefiniowaliśmy.

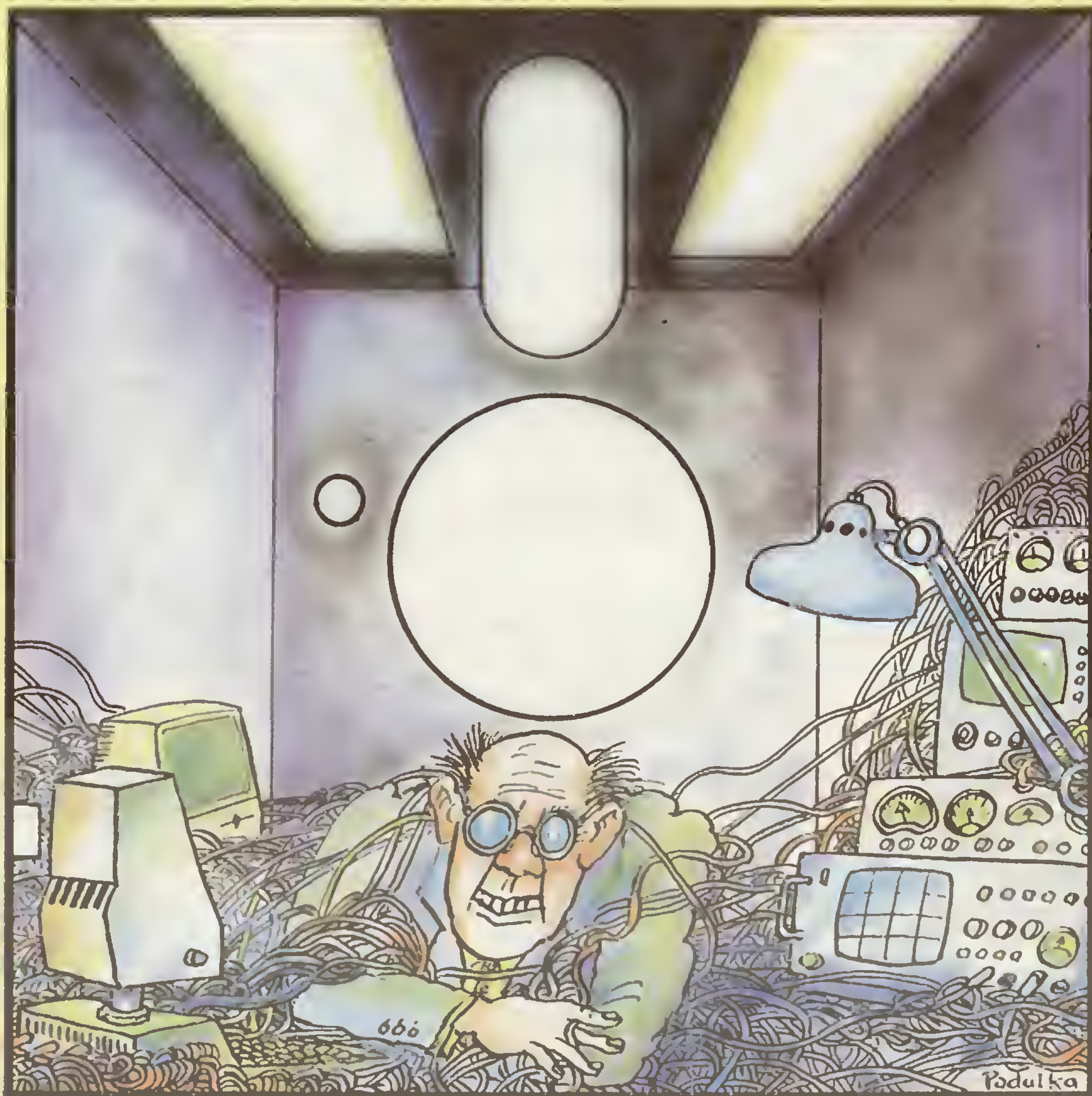
```
to raki
 cs srednie
 linia 1
 k t o . t o . t a k i
 linia 2
 a . t o . r a k i
 linia 3
 t a t a . r a k . i . m a m a . r a k
 linia 4
 a . t a m . m a m y . t a t a r a k
end
```

Życzę cierpliwości podczas definiowania całego alfabetu.

Romek

KONKURS





Czy wiecie jak wygląda dyskietka? Jest to krążek pokryty z obu stron cieniuteńką warstwą materiału magnetycznego. Informacje są zapisywane na koncentrycznych okręgach zwanych ścieżkami, na których mieści się po kilkanaście sektorów. Z dyskietkami należy obchodzić się bardzo ostrożnie. Szkodzi im zbyt wysoka temperatura, brud, kurz, zbyt mocne wyginanie oraz silne pole magnetyczne. Czasem jednak, mimo szczególnej troski, zdarzają się przykre niespodzianki...

Profesor Turbulentny od kilku lat prowadził badania zmierzające do odkrycia nowych materiałów ferromagnetycznych. Zmudne próby dały ostatnio niespodziewanie dobre rezultaty. Profesor zdecydował się zabezpieczyć cenne dokumenty w kasie pancernej zamykanej elektronicznym zamkiem. Szyfr do zamku był zapisany na dyskietce, która niestety została uszkodzona podczas nieostrożnej zabawy z elektromagnesem. Nie pomogło nawet najnowocześniejsze oprogramowanie służące do odzyskiwania utraconego zapisu. Profesor Turbulentny ma zmarnowane święta!

Pomóżcie zrozpaczonemu naukowcowi Fragmenty notatek pozwalają odtworzyć część informacji, które znajdowały się na dyskietce. Posługując się nimi należy zapisać odpowiednie dane, tak jak robiła to głowica stacji dysków, a więc do właściwych ścieżek i sektorów. Odgadnięte słowa wpisujemy zgodnie z kierunkiem ruchu wskazówek zegara rozpoczynając od pola sąsiadującego z kodem ścieżki i sektora. Na przykład dla ścieżki 12 i sektora 4 będzie to kod 124. Hasło otwierające kasę pozwala odczytać umieszczony obok program. Zakradło się do niego kilkanaście błędów. Kolejne numery linii z ukrytymi błędami oznaczają poszczególne litery hasła np. 1027 to 10 ścieżka, 2 sektor, 7 litera słowa.

Jeśli masz kłopoty z przeanalizowaniem programu, to poproś kolegę, który ma komputer. Maszyna pomoże Ci znaleźć błędy. Jeśli wykorzystasz do tego Spectrum, to pamiętaj, że instrukcja RND ma trochę inną postać.

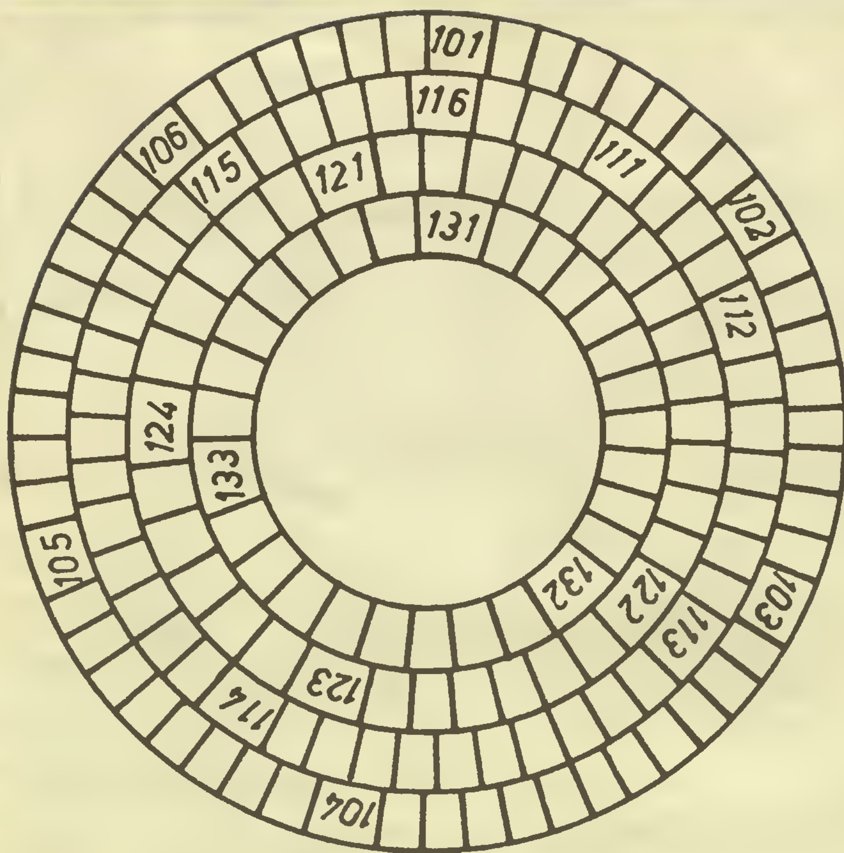
Rozwiązania (hasło otwierające kasę) tylko na kartkach pocztowych z naklejonym kuponem konkursowym prosimy przysyłać do 20 kwietnia 1988 na adres redakcji:

„Bajtek"
*ul. Wspólna **61***
*00-687 **Warszawa***

*Główną nagrodą w naszym konkursie jest komputer Atari 65 XE z magnetofonem ufundowany przez **PZ KAREN***

(b)

```
1010 DIM A$(15):REM IMIE GRACZA
1011 DIM P$(3),REM ODPOWIEDZ
1013 LET PUNKT=0
1014 LET KONIEC=1060:LET P=1
1020 PRINT "CZESC!"
1022 PRINT "JA JESTEM KOMPUTER, A TY J
AK MASZ NA IMIE ";
1023 INPUT A$
1026 LET DL=LEN(A$
1030 GOSUB 1330:PRINT "CZESC ";A$;"!"
1031 PRINT "JESTES FAJN";
1032 IF A$(DL)="A" THEN PRINT "A DZIEW
CZYNA!":LET P=0
1033 IF A$(DL)<>A THEN PRINT "Y CHLOPA
K !"
1040 PRINT "WIESZ ";A$;" CO TO JEST NA
JWIEKSZY WSPOLNY PODZIELNIK ";
1042 INPUT P$
1044 IF P$<>"TAK" AND P$<>"NIE" THEN G
OSUB 1310:GOTO 1040
1046 IF P$="NIE" THEN GOSUB 1330:GOSUB
 1210:GOTO 1040
1048 IF P$="TAK" THEN GOSUB 1330:GOSUB
 1110
1050 PRINT "CZY CHCESZ ";A$;" SPROBOWA
C JESZCZE RAZ ";
1052 INPUT P$
1055 IF P$<>"TAK" AND P$<>"NIE" THEN G
OSUB 1310 GOTO 1050
1058 IF P$="TAK" THEN GOSUB 1111:GOTO
1050
1059 GOSUB 1330
1060 IF P THEN PRINT "ZDOBYLES ";
1061 IF  NOT P THEN PRINT "ZDOBYLAS ";
1062 PRINT PUNKT;" PKT."
1063 PRINT "DZIEKUJE I DO ZOBACZENIA"
1065 STOP
1110 PRINT "NO TO SIE ZARAZ PRZEKONAMY
, CZY GO UMIESZ OBLICZAC!"
1111 LET L1=1+INT(30*RND(1)))
1112 FOR K=1 TO 10*RND(1)
1113 LET L2=1+INT(30*RND(1))
1114 GOSUB 1330
1115 PRINT "SPROBUJ OBLICZYC NWP Z LIC
ZB L1=";L1;" ORAZ L2=";L2
1120 GOSUB 1330
1123 PRINT "JA JUZ DAWNO SKONCZYLEM, A
 TY ";A$
1125 GOSUB 1230
1126 LET B=0:REM LICZNIK BLEDOW
1127 PRINT "PODAJ JAKI MASZ WYNIK ";
1130 INPUT W:REM GRAJACY PODAJE WYNIK
1133 IF W<>A THEN GOSUB 1145:GOTO 1127
1135 GOSUB 1150
1139 RETURN
1145 LET B=B+1:IF B 3 THEN PRINT "BARD
ZO ";:ENDIF
1146 PRINT "ZLE! MUSISZ POLICZYC JESZC
ZE RAZ!"
1148 RETURN
```



ATARI • ATARI • ATARI

NOTATKI PROFESORA TURBULENTNEGO

ŚCIEŻKA 10:

SEKTOR 1: Twórca oszałamiających sukcesów ATARI i Commodore

SEKTOR 2: Precyzyjny opis przedstawiający sposób rozwiązania problemu.

SEKTOR 3: Program, który podczas działania pobiera instrukcję, rozpoznaje jej znaczenie, a następnie wykonuje ją bez tworzenia kodu wynikowego.

SEKTOR 4: Inaczej oprogramowanie.

SEKTOR 5: Stałe, zmienne i funkcje połączone operatorami działań arytmetycznych i logicznych np. mnozenia, dzielenia, odejmowania lub dodawania.

SEKTOR 6: Program umożliwiający redagowanie tekstu.

ŚCIEŻKA 11

SEKTOR 1. Podstawowa jednostka informacji odpowiadająca wystąpieniu jednego z dwóch możliwych stanów

SEKTOR 2: Znany matematyk, fizyk, pisarz i filozof francuski z XVII w: Nazwa jednego z języków programowania.

SEKTOR 3: Proces tłumaczenia z języka zrozumiałego dla użytkownika na jęzvk maszyny.

SEKTOR 4: Układ scalony o wielkiej skali integracji, jeden z najwazniejszych w mikrokomputerze.

SEKTOR 5: Wszystkie informacje przesyłane do i wychodzące z komputera

SEKTOR 6: Pamięć umożliwiająca zapis i odczyt informacji.

ŚCIEŻKA 12

SEKTOR 1: Cecha umożliwiająca przenoszenie oprogramowania z jednego komputera na inny; inaczej zgodność.

SEKTOR 2: Urządzenie służące do sterowania ruchem kursora, spełniająca podobną funkcję jak joy stick.

SEKTOR 3: Pierwsza całkowicie elektroniczna maszyna cyfrowa skonstruowana w 1946 roku.

SEKTOR 4 Jeden z rodzajów pamięci stałej umożliwiający skasowanie zapisu przez naświetlenie promieniami ultrafioletowymi.

ŚCIEZKA 13

SEKTOR 1: Podstawowe urządzenie umożliwiające wprowadzenie informacji do komputera.

SEKTOR 2 Symbol odpowiadający rozkazom, ułatwiający zapamiętanie ich znaczenia

SEKTOR 3 Strukturalny język programowania dla komputerów Atari.

```
1150 IF B=0 THEN PRINT "BARDZO ";
1151 PRINT DOBRZE, WYNIK PRAWIDLOWY!"
1152 LET PUNKT=PUNKT+1
1155 RETURN
1210 PRINT "NAJWIEKSZY WSPOLNY PODZIEL
NIK DWOCH LICZB CALKOWITYCH L1 I L2 ";
1211 PRINT "JEST TO NAJWIEKSZA LICZBA.
 PRZEZ KTORA MOZNA JE OBIE PODZIELIC"
1213 GOSUB 1330
1215 PRINT "POKAZE TO NA PRZYKLADZIE:"
1217 PRINT "PODAJ PIERWSZA LICZBE L1
";:INPUT L1
1219 PRINT "A TERAZ DRUGA ";
1220 INPUT L2
1223 GOSUB 1230:GOSUB 1330
1225 PRINT "NWP LICZB L1=";L1;" I L2="
L2;" JEST ";A
1227 GOSUB 1330
1228 PRINT "TERAZ JUZ NA PEWNO ";:RETU
RN
1231 LET A=L1:LET B=L2
1233 IF A<B THEN LET T=B:LET B=A:LET A
=T
1235 LET R=A-(INT(A/B))*B
1237 LET Q=INT(A/B)
12   LET A=B LET B=R
1243 IF R   THEN GOTO 1235
1245 RETURN
1310 PRINT "NIE ROZUMIEM!"
1315 IF P THEN PRINT "POWINIENES ";
1317 IF  NO P THEN PRINT "POWINNAS ";
1320 PRINT "ODPOWIEDZIEC TAK LUB NIE""
1325 PRINT "SPROBUJ JESZCZE RAZ!"
1328 RETURN 1227
1330 FOR K=1 TO 10 STEP 5
1332 FOR L=4 TO 30:PRINT "*";
1334 NEXT K
1335 PRINT
1337 NEXT L
1339 RETURN
```

NIE TYLKO KOMPUTERY

CYFROWE WYZWANIE

dokończenie ze str. 32

nazwą Beta-max. Był i jest znacznie doskonelszy, zapewniający lepszą jakość wizji i fonii, niż rozpowszechniony obecnie VHS. Cóż, kiedy tajemnice techniczne Bety-max Sony pragnął zachować wyłącznie dla siebie. Tymczasem najpoważniejszy rywal — JVC — zaprojektował nieco mniej zaawansowany technologicznie system, którym z iście japońskim sprytem zaczął wypierać Sony z rynku. Trik JVC był tylko do pewnego stopnia samobójczy — przekazanie czyhającym tylko na taki moment głodnym wilkom branży elektronicznej sekretow własnego systemu VHS. Prawie momentalnie w sklepach zaroiło się od magnetowidów firm wszelakich, naturalnie tych oznaczonych trzema literkami. Beta-max i Sony zepchnięci zostali na margines.

Kiedy fala wideo dotarła do Polski było już po wszystkim. Kaseta magentowidowa i magnetowid kojarzyły się wyłącznie z systemem VHS. Co prawda byli i tacy, którzy widzieli Betę-max marki Sony, ale któż wchodziłby z nimi w układy skoro posiadali sprzęt... nietypowy, uniemożliwiający wymianę nagrań. Na domiar złego dla Sony na kasecie VHS można zmieścic cztery godziny programu, tymczasem Beta-maxy z trudnością grały ponad godzinę. Najnowsze taśmy wyprodukowane eksperymentalnie przez walczący jak lew koncern Sony mieszczą pięc godzin, ale konkurencja zrewanżowała się niewiarygodnie cienkimi... ośmiogodzinnymi VHS.

To, co skazało u nas na ostracyzm właściciell Beta-maxow zawładnęło również wideobiznesem na Zachodzie. W wypożyczalniach dostac można co prawda kasety systemu lansowanego przez Sony, ale jest ich coraz mniej. Najnowsze filmy z rzadka utrwala się na Betamaxach. Kiedy po raz pierwszy wylała rzeka VHS, Sony stracił aż 40 proc. zdawałoby się dobrze opanowanego rynku magnetowidowego. W roku ubiegłym sprzedano w Stanach Zjednoczonych całe morze nowych magnetowidów o wartości 5.25 mld dolarów. Tylko 10 proc. nabywców zdecydowało się na inne systemy niż VHS. Liczbę posiadaczy Beta-maxów w USA ocenia się dziś na 20 milionów. Obecnie mogą z czystym sumieniem powiedzieć sobie, że zostali zdradzeni... Sony bowiem od stycznia 1988 roku rozpoczął sprzedaż magnetowidów VHS w Stanach Zjednoczonych. Co prawda rzecznicy koncernu powtarzają bez ustanku, iż nadal produkowane będą urządzenia pozwalające na używanie tych pierwszych, mniejszych kaset, jednak tak, czy inaczej, jest to koniec systemu Beta-max. Kapitulacja i przyznanie się do porażki.

Cyfrowe wyzwanie dźwiękowe postawiło przed Sony kolejny problem do rozstrzygnięcia. Gdyby można było podpowiedzieć skośnookim specom od amerykanskiej klienteli, trzeba byłoby im opowiedzieć o początkach kina w Stanach Zjednoczonych. O tym, z jakim uporem Tomasz Edison i jego koncern bronili swego monopolu na produkcję ruchomych obrazów, a także o tych, którzy w ucieczce przed jego bojówkami, operującymi na Wschodnim Wybrzeżu wbrew przepisom kręcąc korbkami pierwszych kamer, trafili na cudowne, spokojne miejsce — Hollywood.

Wojciech Łuczak

CYFROWE WYZWANIE

Wszystko wygląda podobnie. I obudowa i sam układ. Właściwie na pierwszy rzut oka maszyneria nie różni się specjalnie od najnowszych modeli magnetofonów, nazwijmy je konwencjonalnymi. Bo w gruncie rzeczy od roku Japończycy, a od kilku tygodni Amerykanie mają do czynienia z trochę tylko innym urządzeniem do zapisywania i odtwarzania dźwięku na przesuwającej się taśmie. Ale sposób owej rejestracji, czy odkodowywania głosów w absolutnej nowince technologicznej ostatnich czasów — magnetofonie cyfrowym, czyni go doskonalszym od każdego dotychchczasowego rodzaju sprzętu, jaki skonstruowano ku wygodzie i uciesze fonoamatorów.



SYSTEM BETA-MAX BYŁ I JEST ZNACZNIE DOSKONALSZY NIŻ ROZPOWSZECHNIANY OBECNIE VHS

Kompaktowe płyty i laserowe odtwarzaczę nie szokują już dziś nikogo, nawet w Polsce. Dźwięk rozłożony na czynniki pierwsze i zaszyfrowany w postaci kombinacji zerojedynkowych, dotrzec może do naszych uszu bez kłopotliwego pośrednictwa igły drgającej w rowku wyżłobionym w czarnym, wirującym krążku. W epoce kompaktów hasło „szum starej płyty" stać się powinno zabytkiem językowym. Czy może szumieć coś, czego w sensie fizycznym nie dotyka żaden mechanizm, a jedynie muska promień lasera?

Podobnie czyste tony, głosy z niesłychaną precyzją oddające ciepło i nasycenie dźwięku, zapewnia magnetofon cyfrowy. Działa prawie jak kompakt. Choć szczegóły techniczne osłonięte są, jak zawsze w przypadku nowości, zrozumiałą tajemnicą, można powiedzieć, jak pisze amerykański magazyn „Time", że „... komputerowe mikrochipy rozbijają głosy na miliardy bitów informacji, które są następnie magazynowane na taśmie magnetycznej...

Naturalnie byłoby zbyt pięknie, gdyby nośnikiem informacji o nagranych dźwiękach była zwykła kaseta magnetofonowa.

Ta nowa — DAT (digital audio tape) jest mniejsza i pojemniejsza od tradycyjnej. Na muzyczną scenę wchodzi jednak z oporami, tak jak i cyfrowy magnetofon. Technologia DAT jest bowiem czymś w rodzaju potężnego wyzwania dla władców imperium płyty kompaktowej. A że często o rozwoju obu technik odtwarzania cyfrowego decydują ci sami ludzie, w tak zwanej „branży" dzieją się cuda, o które nikt w Polsce nie podejrzewałby ludzi amerykańskiego biznesu.

Yukinori Ishikawa — tokijski korespondent tygodnika „Time" doniósł, iż Japończycy z umiarkowanym zapałem przyjęli magnetofon cyfrowy. Choć już od roku w sklepach Nipponu oferuje się ten najwyższej jakości sprzęt, spodziewanego boomu nie odnotowano. Mieszkańcy Kraju Kwitnącej Wiśni nie są bowiem aż takimi szaleńcami konsumpcji aby rzucać się bez opamiętania na każdą nowinkę. Szturmu klientów oczekuje się natomiast w Stanach Zjednoczonych. Styczniowa wystawa elektroniki dla każdego w Las Vegas dowiodła, że w Ameryce rozpoczyna się prawdziwa cyfrowa rewolucja skierowana przeciwko dotychczasowemu stylowi nagrywania i odtwarzania.

Praktycznie każda licząca się na rynku amerykańskim firma pokazała swój model magnetofonu cyfrowego. Przy czym sercem wszystkich jest zoptymalizowana już kaseta DAT. Pół tuzina wytwórców zapewnia, że ich sprzęt pojawi się w sklepach już latem 1988 roku. Forpocztę natarcia tworzą spółki związane z przemysłem motoryzacyjnym. Od lutego konsorcjum Kenwood oferuje nabywcom swój „set" — radio połączone z cyfrowym odtwarzaczem kaset do zamontowania w samochodzie. Ford obiecuje wyposażyć w cyfrową aparaturę Sony swoje Lincolny Continentale wypuszczane od czerwca. Firmy Harman/Kardon, Marantz i Casio nie ograniczają się wszakże do sprzętu odczytującego nagrane fabrycznie DAT, a tych nie jest jeszcze tak dużo (mówi się w Las Vegas o uruchomieniu produkcji 100 pierwszych tytułów z muzyką poważną i jazzem). Rzucają bowiem latem pierwsze prawdziwe magnetofony cyfrowe.

Za nowości jeszcze nie nazbyt upowszechnione trzeba oczywiście sporo zapłacić. Odtwarzacz taśmowy Kenwooda kosztować ma około 2000 dolarów. Najtańszy i najprostszy magnetofon Casio ceni się na 1000 dolarów.

Fachowcy twierdzą, że ten dość wysoki pułap można będzie obniżyć, kiedy do cyfrowego amerykańskiego wyścigu włączą się tacy potentaci jak Sony. Ale w przypadku tej firmy mamy do czynienia z klasycznym syndromem giganta dławiącego się swoją zdobyczą...

Koncern Sony mógłby przecież w ciągu kilku miesięcy zalać Stany Zjednoczone swymi cyfrowymi cudeńkami, a przy tym skalkulować cenę na poziomie nie pozwalającym ruszyć palcem konkurencji już istniejącej i tej potencjalnej. Czemu się waha? Czemu zwleka?

Problem jest bardzo skomplikowany, choć ma wszelkie pozory prostoty. U podstaw pytania co dalej — leży następująca konstatacja — każdy właściciel cyfrowego magnetofonu jest w stanie utrwalić na taśmie głos swego ulubionego wykonawcy w sposób nie gorszy niż na płycie kompaktowej — pożyczając ją, a następnie reprodukując. W technice cyfrowej nie istnieje bowiem kwestia zniekształcania jakości nagrania przy kolejnej reprodukcji. Producenci płyt kompaktowych — do niedawna monopoliści czystych głosów i fantastycznej jakości dźwięku mogą więc ze spokojem zacząć pakować manatki i szukać tablicy z napisem „bankructwo".

Szkopuł w tym, iż w styczniu 1988 roku koncern Sony wchłonął firmę CBS Records — absolutnego potentata płytowego Stanów Zjednoczonych. Giganta posiadającego w swej stajni takie nazwiska jak Bruce Springsteen, czy Michael Jackson. Goliata, który jednak przestraszył się dawidowych rozmiarów magnetofonu cyfrowego. Szefowie CBS od dawna należą do czołówki tych wytwórców kompaktów, którzy wypowiedzieli bezwzględną wojnę magnetofonom cyfrowym bojąc się, jak końca świata, utraty absolutnej władzy nad cyfrowym królestwem. Niektórzy mniejsi fabrykanci magnetofonów skłonni są do pewnych ustępstw — pisze się w USA o wprowadzeniu „elektronicznej zapadki" zwanej Solo umożliwiającej skopiowanie płyty kompaktowej, wykluczającej jednak dalsze kopiowanie nagranej taśmy. Lobby płytowe wszakże nie ustępuje i żąda od... Kongresu Stanów Zjednoczonych nałożenia na konsorcja posługujące się kasetą DAT obowiązku zaopatrzenia ich sprzętu w mechanizm uniemożliwiający przegrywanie jakiejkolwiek muzyki wcześniej zarejestrowanej fabrycznie w systemie cyfrowym. Przepychanka trwa, a Sony traci miliony...

Nie po raz pierwszy na skutek błędnej decyzji o fatalnych skutkach. Trzynaście lat temu japoński trust zwalił z nóg konkurencję swoją nadzwyczajną nowością — magnetowidem. Oryginalny system zapisu dźwięku i obrazu opatrzono